UNIVERSITÉ PARIS 13,
PARIS SORBONNE CITÉ

Laboratoire d'Informatique de Paris-Nord (LIPN)

THESIS

presented by
Nazanin FIROOZEH

for the degree of
DOCTOR OF PHILOSOPHY

Semantic-oriented Recommendation for Content Enrichment

Recommandation sémantique pour l'enrichissement du contenus textuels

≪CONFIDENTIAL THESIS≫

Doctoral committee:

Béatrice DAILLE	Professor	University of Nantes	Reviewer
Mohand BOUGHANEM	Professor	Paul Sabatier University	Reviewer
Massih-Reza AMINI	Professor	Grenoble Alpes University	Examiner
Davide BUSCALDI	Associate Professor	University Paris 13	Examiner
Adeline NAZARENKO	Professor	University Paris 13	Supervisor
Fabrice ALIZON	Engineer (PhD)	Pixalione SAS	Co-Advisor

Abstract

In this thesis, we aim at enriching the content of an unstructured document with respect to a domain of interest. The goal is to minimize the vocabulary and informational gap between the document and the domain. Such an enrichment which is based on Natural Language Processing and Information Retrieval technologies has several applications. As an example, filling in the gap between a scientific paper and a collection of highly cited papers in a domain helps the paper to be better acknowledged by the community that refers to that collection. Another example is to fill in the gap between a web page and the usual keywords of visitors that are interested in a given domain so as it is better indexed and referred to in that domain, *i.e.* more accessible for those visitors.

We propose a method to fill that gap. We first generate an enrichment collection, which consists of the important documents related to the domain of interest. The main information of the enrichment collection is then extracted, disambiguated and proposed to a user, who performs the enrichment. This is achieved by decomposing the problem into two main components of keyword extraction and topic detection. We present a comprehensive study over different approaches of each component. Using our findings, we propose approaches for extracting keywords from web pages, detecting their underlying topics, disambiguating them and returning the ones related to the domain of interest. The enrichment is performed by recommending discriminative sets of semantically relevant keywords, *i.e.* topics, to a user. The topics are labeled with representative keywords and have a level of granularity that is easily interpretable. Topic keywords are ranked by importance. This helps to control the length of the document, which needs to be enriched, by targeting the most important keywords of each topic. Our approach is robust to the noise in web pages. It is also knowledge-poor and domain-independent. It, however, exploits search engines for generating the required data but is optimized in the number of requests sent to them. In addition, the approach is easily tunable to different languages. We have implemented the keyword extraction approach in 12 languages and four of them have been tested over various domains. The topic detection approach has been implemented and tested on English and French. However, it is on French language that the approaches have been tested on a large scale: the keyword extraction on roughly 400 domains and the topic detection on 80 domains.

To evaluate the performance of our enrichment approach, we focused on French and we performed different experiments on the proposed keyword extraction and topic detection methods. To evaluate their robustness, we studied them on 10 topically diverse domains. Results were evaluated through both user-based evaluations on a real application context and by comparing with baseline approaches. Our results on the keyword extraction approach showed that the statistical features are not adequate for capturing words importance within a web page. In addition, we found our proposed approach of keyword extraction to be effective when applied on real applications. The evaluations on the topic detection approach also showed that it can effectively filter out the keywords which are not related to a target domain and that it labels the topics with representative and discriminative keywords. In addition, the approach achieved a high precision in preserving the semantic consistency of the keywords within each topic. We showed that our approach outperforms a baseline approach, since the widely-used co-occurrence feature between keywords is not

enough for capturing their semantic similarity and consequently for detecting semantically consistent topics.

Résumé

Dans cette thèse, nous cherchons à enrichir le contenu d'un document non structuré par rapport à un domaine d'intérêt en nous appuyant sur des techniques de traitement du langage naturel (TAL) et de recherche d'information. L'objectif est de minimiser l'écart lexical et informationnel susceptible d'exister entre le document et le domaine considérés. Il peut s'agir, par exemple, de combler le fossé qu'il peut y avoir entre un article scientifique et une collection d'articles de TAL fréquemment cités, en sorte que l'article soit mieux reconnu par la communauté du TAL. On peut aussi chercher à combler l'écart entre une page web et les mots-clefs des visiteurs qui s'intéressent à un domaine donné, afin que la page soit mieux indexée et référencée dans le domaine considéré, c'est-à-dire plus facile d'accès pour ces visiteurs.

Nous proposons une méthode pour réduire cet «écart sémantique». Nous générons d'abord une collection d'enrichissement rassemblant des documents importants liés au domaine d'intérêt. Cette collection est analysée pour en extraire les principaux éléments d'information qui sont désambiguïsés et proposés à l'utilisateur en charge de l'enrichissement. Nous avons décomposé ce problème en deux parties principales : l'extraction de mots-clés et la détection des principaux thèmes. A partir de l'analyse des méthodes existantes dans ces deux domaines, nous proposons une nouvelle approche permettant d'extraire des mots-clés à partir de pages web, de détecter leurs thèmes sous-jacents, de les désambiguïser et de retourner à l'utilisateur ceux qui semblent liés à son domaine d'intérêt. L'enrichissement est assuré par l'utilisateur à partir des thèmes (*topics*) qui lui sont proposés, ceux-ci étant représentés par des ensembles discriminants de mots-clés sémantiquement pertinents. Ces thèmes sont étiquetés avec des mots-clés représentatifs et ont un niveau de granularité qui les rend interprétables. Les mots-clés des thèmes sont classés par importance, ce qui permet de contrôler la longueur du document enrichi, en ciblant les mots-clés les plus importants dans chaque thème.

Notre approche est robuste au bruit présent dans les pages web. Elle est également pauvre en connaissances et indépendante du domaine considéré. Elle exploite les moteurs de recherche pour générer les données requises mais en optimisant le nombre de requêtes qui sont envoyées aux moteurs. En outre, l'approche peut être facilement adaptée à différentes langues. Nous avons implémenté l'extraction des mots-clés pour 12 langues et quatre d'entre elles ont été testées sur des domaines variés. La détection des thèmes a été mise en œuvre et testée en anglais et en français. L'ensemble de la méthode a été testée à grande échelle en français, sur 400 domaines pour l'extraction de mots-clés et 80 domaines pour la détection de thèmes.

Pour évaluer la performance de notre méthode d'enrichissement, nous nous sommes concentrée sur le français et nous avons effectué différentes expériences d'extraction de mots-clés et de détection de thèmes. Pour évaluer leur robustesse, nous avons appliqué nos méthodes sur 10 domaines thématiquement variés. Les résultats ont été évalués par des utilisateurs dans un contexte applicatif réel et par comparaison avec des approches de référence. Les résultats montrent que notre approche d'extraction de mots-clés fonctionne mieux que celles qui reposent uniquement sur des caractéristiques statistiques, ces dernières capturant imparfaitement l'importance des mots dans une page web.

Sur la détection de thèmes, les évaluations ont également montré que notre approche permet de filtrer les mots-clés qui ne sont pas liés au domaine cible et à étiqueter les thèmes avec des mots-clés représentatifs et discriminants. On observe en outre une bonne précision dans les résultats et une bonne cohérence sémantique au sein de chaque thème. Nous avons montré que notre approche surpasse une approche de référence reposant sur la cooccurrence entre mots-clés, laquelle rend imparfaitement compte de la similarité sémantique entre les mots-clefs et ne parvient pas à construire des thèmes sémantiquement cohérents.

Acknowledgements

Firstly, I would like to express my gratitude to my supervisor Prof. Adeline Nazarenko for her academic support during my PhD work. In particular, I want to thank her for supporting me during the hard times and providing help when I most needed it.

I would like to express my special appreciation and thanks to Dr. Fabrice Alizon, the chief executive officer of Pixalione, who has been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a researcher.

My gratitude also goes to the rest of my thesis committee, Prof. Béatrice Daille, Prof. Mohand Boughanem, Prof. Massih-Reza Amini, and Dr. Davide Buscaldi, for their insightful comments and suggestions and for their participation in my thesis committee.

I thank all my colleagues and friends in PixalioneLab for our stimulating discussions and collaborations and for all the fun we have had in the last four years. In particular, I thank Marwa Ghorbel who collaborated as an intern in some steps of the project.

Some special words of gratitude go to my friends. I would like to specifically thank Moein Montazeri for his presence in the difficult periods and for his support.

Last but not least, thanks to my family for their unbelievable kindness and support. Words cannot express how grateful I am to my parents and my sisters. I am sorry that my father has not lived to see the completion of my work. His unconditional love will always encourage me in my life and I dedicate this thesis to his memory.

Contents

List of Figures

List of Tables

Part I

General introduction

Chapter 1

Introduction

Contents

Nowadays many textual documents are generated daily by different people all over the world. Depending on the expertise of the person who writes the content of a document with respect to a domain, the quality of the document could vary. Some documents better cover the domain of study: they use representative and domain-specific vocabulary and contain pieces of information, which effectively relate the document to the domain. On the contrary, some documents may poorly cover the domain of study. They provide less information for readers and affect the performance of Information Retrieval (IR) methods and Natural Language Processing (NLP) tools.

Enriching a document with respect to a domain enhances the quality of that document. The enrichment can be performed by adding pieces of text to the content of the document or assigning metadata to it. The former type of enrichment requires language modeling approaches for generating text. In the state of the art of NLP, the latter type of enrichment has been mostly studied. Stajner et al. (2010) make use of the content of a document to generate metadata for it. Some researchers, however, exploit external resources for enriching a document. Examples are Hotho et al. (2003) and Hu et al. (2008), who respectively use WordNet and Wikipedia to improve document clustering. The enrichment can be performed from a vocabulary point of view in order to enrich the vocabulary of a document with respect to a source of information or from an informational point of view, where pieces of information that are not covered by the document are added to its content.

More specifically, considering a document and a source of information, quite often, there is a gap between them, which is referred to as "semantic gap" in this thesis (Figure 1.1). Depending on the target application, the source of information can be a term (query), a collection of documents or more generally, a domain of interest. The semantic gap can be of two types:

- Vocabulary-based: In this case, the gap is related to the chosen vocabulary, in which the input document and the source of information deal with the same topic but do not use the same words, which make them difficult to relate. As an example,

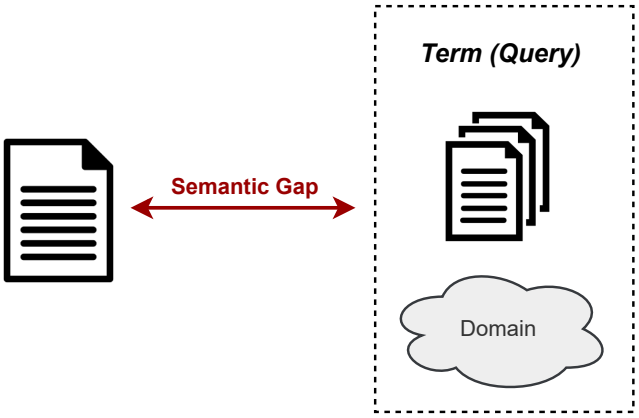


Figure 1.1: Semantic gap between a document and a source of information

two scientific papers originated from two distinct communities may have different vocabularies. This is also a typical issue in search applications, where the vocabulary of a user who specifies the query is not the same as the vocabulary of documents. In this case, direct information retrieval methods fail to capture the relation between the query and the documents and consequently cannot effectively detect the documents which are relevant to the query.

- Informational: In this case, the gap is related to the pieces of information that a document and a source of information contain. More specifically, in this type of gap, the document and the source of information deal with connected or overlapping topics but one is missing part of the information of the other. For instance, one can ask what information is missing for a scientific paper on graphs to be referred by the NLP community. Another example is the work by Guo et al. (2013), which targets the informational gap and augments the context of tweets using the information obtained from news contents. The given example in their work is "Pray for Mali" tweet, which does not explicitly cover the information about the "war" and "French army participation" events.

Filling in the vocabulary-based and the informational gap between a document and a source of information is essential, since it enables the NLP tools to better understand the document and as a result provides richer context for NLP tasks like clustering, searching or other text analysis applications.

The missing vocabulary or information in a document has different levels of importance: some are more critical than others. Since the length of the input document must be reasonable after the enrichment procedure, one may need to target only the vocabulary units that are more common or the pieces of information that are more important in the domain. Prioritizing the vocabulary and the pieces of information and targeting the most important ones is becoming vital in competitive environments, where documents with richer contents tend to be visited by more people. As an example, we can refer to

the competitive environment of the web, where different pages are competing for a higher position and more visibility in search results.[1]

Filling in the semantic gap with manual analysis is very expensive and not feasible. Hence, there is a need for tools to understand the input document and also the studied source of information so as to return the relevant vocabularies or pieces of information. These tools must be able to handle the ambiguity problem while performing the analysis. For instance, in case of using dictionaries as an external resource for filling in the vocabulary-based semantic gap, not all the synonyms of a word in a dictionary may have close semantic relation with the input document.

In the following sections, we explain in more details the problem that we study in this thesis. We then discuss the research questions that we aim at answering and present the main objectives of our work and the properties of our approach.

1.1 Problem statement

In this thesis, we focus on enriching the content of web pages, which are one type of unstructured documents. More specifically, we focus on web pages with non-streaming data, such as commercial web pages, rather than streaming ones, such as news. The enrichment is performed with respect to a domain of interest in order to fill in the semantic gap between an input document and the target domain. The main focus of this thesis is on filling in the informational gap, even though the vocabulary gap can be filled in as well. By filling in the semantic gap, more information about the domain is given to visitors of web pages and richer context is provided for text analysis tools. As an example, web pages can be better indexed by search engines, which results in higher visibility on the web. They can be also categorized more effectively when there is enough contextual information for the categorization task.

Our enrichment application involves a user, who performs the enrichment procedure. It is modeled as the following (Figure 1.2): the user targets a web page as an *input document* that is aimed to be enriched with respect to a domain, which specifies the enrichment's point of view. The domain is labeled with a representative keyword, which is also determined by the user. Depending on the closeness of the domain of study and the input document, the semantic gap between them could be bigger or smaller. If they are close, the gap is small and mostly vocabulary-based. However, if they differ, the semantic gap is probably more informational and the goal would be to enrich the input document with the information retrieved from the domain of study.

The user also interacts with the system at the end of the enrichment procedure, when the detected missing vocabulary or information, called *enrichment information*, is recommended to the user. The recommended information is as a set of keywords, which can be considered as the metadata for the input document. Although this metadata can be

[1] In addition to content, the link structure also affects the ranking of web pages. However, since the link structure is out of scope of this work, we do not discuss it here.

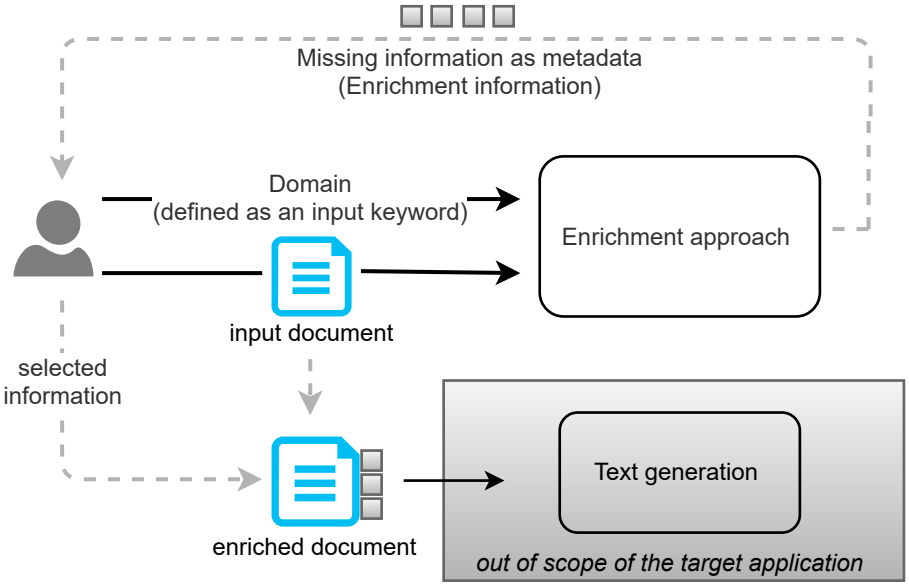


Figure 1.2: Flowchart of the target enrichment application

further used in a language generation process in order to generate pieces of text for the input document, we do not focus on this step in this thesis and merely recommend a set of keywords to the user. The user then makes the final decision for adding the desired information to the input document. Hence, the detected information is not added directly to the document. It is firstly verified by the user. This manual verification is required in order to avoid adding pieces of information that the user does not find relevant to the input document. As an example, information about "TV Mounts" is related to the domain of "TV" but is nevertheless not relevant to be added to a commercial web page, which merely sells "TV". Hence, such information is filtered out by the user and is not added to the input document.

While enriching the content of an input document, its length must remain reasonable. Very long documents are not interesting for readers and they may cause loading issues in some applications. Recommending a large amount of information to users also makes the enrichment application complex for them. Due to these restrictions, it is almost impossible to fill in the whole semantic gap between a document and the studied domain. Hence, we need to target the most common vocabulary or the most critical information within the domain of study to eventually recommend limited but significant enrichment information to users and to "minimize" the semantic gap. Considering a collection of documents as representative of the domain of study, two steps need to be performed in order to obtain the significant and frequently discussed information of the domain:

1. Building an *enrichment collection* that is a collection of important and representative documents in the studied domain,

2. Targeting the significant information in the enrichment collection.

The first step requires an approach for identifying the most important documents within a great number of available documents in a domain. In the second step, an automatic approach of extraction is needed to analyze the content of each document and to return the main information that it discusses.

In this thesis, we address two research questions related to the second step:

1. How to extract the main information of documents?
2. In case of having multi-topic documents, how to distinguish the information in different topics?

To extract the information of documents, there are different criteria to be taken into account. More specifically, we need to study the properties of the required information and to use an effective approach for extracting this information from documents.

Although some documents discuss only one topic, some others are composed of different topics. As an example, it is very likely that a commercial web page discusses various topics, such as different types of products, user reviews, payment policies, etc. A scientific paper also contains several topics, including a research topic, affiliations and acknowledgment. Having such multi-topic documents, the extracted information also belongs to different topics. However, not all these topics are related to the studied domain. Enriching the content of the input document using topically irrelevant information would decrease its quality and obviously would not be a correct way of enrichment. Therefore, an approach is required to distinguish the different topics, to select the ones which are related to the domain of study and to further enrich the input document by adding the domain-specific information.

It should be noted that although we specifically focus on web pages, the target enrichment problem in this thesis could have applications on other types of unstructured documents. As an example, while writing a scientific paper in a specific domain, highly cited documents in the domain could be analyzed in order to find the information that the input document needs to cover so that to be acknowledged by the community that refers to the highly cited documents. Another example is in writing the news content, where different sources of news could be analyzed in order to detect the related events and to cover them all in the target news content.

In the following, we explain how we target the mentioned challenges and how the research questions are answered in this work.

1.2 Research objective

The aim of this thesis is to perform document enrichment by automatically minimizing the vocabulary and informational semantic gap between a document and a domain of study. We specifically focus on web pages in this thesis and so propose an approach for enriching

the content of web pages. Unlike some works which exploit knowledge bases for enriching a document, we aim at proposing a knowledge-poor approach, which can be applied on any domain.

To perform the enrichment, we initially generate an enrichment collection, which consists of a limited number of important documents within the domain of study. In this step, we rely on the results returned by search engines, as they can effectively detect the important documents within a collection.

To answer the research questions, presented in Section 1.1, we decompose the problem into two main components, each of which answers one of the research questions: *Keyword Extraction* and *Topic Detection*.

In keyword extraction, we automatically extract the information of each document of the enrichment collection. This information is represented as a ranked list of "key" words or phrases, which are considered as representatives of the documents. We are interested in extracting both single and multi-token keywords. However, to avoid extracting too specific keywords, we limit the length of the extracted keywords in terms of the number of the constituent tokens.

To overcome the problem of extracting multi-topic information (keywords), we propose a topic detection approach, which takes the extracted keywords as input and returns their latent topics as output. In this work, we define a "topic" as a set of keywords with close semantic relationships. The detected topics may, however, not be all relevant to the domain of study and the topic detection approach needs to identify the relevant topics for the enrichment task. Those topics are then considered as the enrichment information that is recommended to a user, who performs the enrichment. Hence, in our approach, due to the ambiguity issues and also to help users to better understand the enrichment information, we recommend it as a set of topics. These topics should have a right level of granularity to be easy for users to interpret.

Pages on the web could be noisy: while writing the content of web pages, people may not follow the same standard as in other types of documents, such as scientific papers; web pages may contain typographical errors; some web pages might be spam pages with uninformative content or unreliable information, etc. The proposed enrichment approach and more specifically, the proposed keyword extraction and topic detection approaches must be robust to the noise in web pages.

In this work, we make use of the context returned by search engines in order to capture the similarity between keywords while detecting the latent topics. To further control the amount of information recommended to users and consequently to control the length of the input document, keywords of each topic are recommended in order of importance. The highest ranked keyword of a topic is considered as its label. This label makes the topic easier and faster for the user to interpret.

In a nutshell, our enrichment application requires to:

- be applicable on web pages as a type of unstructured documents. Hence, it must be robust to the noise in web pages;
- interact with a user, who performs the enrichment procedure;
- be domain-independent;
- be easily tunable to different languages.

Considering these requirements, we propose an enrichment approach with the following properties:

- The enrichment is performed with respect to a domain (point of view).
- The detected enrichment information is recommended to users as sets of semantically related keywords, *i.e.* topics, with a good level of granularity to make the topics easy to interpret for users.
- The approach has a rather light procedure to be executed in a reasonable time, *i.e.* less than half an hour. Due to the interactions with users, the execution time should not be very long even if the approach is not aimed to interact with them in a real time.
- The precision of the recommended information is effective enough, which makes the approach easier and less demanding for users to exploit.
- The enrichment approach is knowledge-poor but it depends on search engines' results. However, search engines are used for generating the required context and the approach is not relying on functionalities of search engines which may change over time. Moreover, the approach is optimized in the number of requests sent to search engines.
- The approach makes a balance between reducing the semantic gap and the length of the enriched document through recommending a limited amount of information as a ranked list of keywords.

1.3 Motivation

As will be explained in the following chapters, there are different approaches to extract keywords from unstructured documents and to disclose the underlying topics in a collection of words or keywords. Nevertheless, most of the keyword extraction approaches in the literature aim at extracting keywords from documents without distinguishing the different topics in the collection of the extracted keywords. In addition, the existing topic detection approaches mainly perform very basic keyword extraction approaches and as a result the keywords within their detected topics may not have a high quality. As an example, Sayyadi

and Raschid (2013) perform both keyword extraction and topic detection but their keyword extraction approach simply consists in extracting all words, noun phrases and named entities from documents. We, however, did not find such basic keywords to be effective enough for our enrichment application.

Some advanced approaches of keyword extraction require a deep analysis of documents, which could make the whole enrichment procedure complex. Some approaches also exploit a corpus of documents for extracting keywords of a single document. This corpus may, however, not be always available. There are approaches of keyword extraction that exploit external knowledge. Due to this property, they are applicable only on a limited number of domains and cannot be considered as domain-independent approaches. In this work, we are interested to propose a simple but effective approach for extracting keywords from web pages. More specifically, we aim at proposing a domain-independent approach that is easily tunable to different languages. We also focus on extracting both single and multi-token keywords to retrieve more information from documents.

The existing topic detection approaches mainly make use of the notion of co-occurrence between keywords in a specific corpus. The co-occurrence between two keywords, however, does not necessarily imply their semantic relatedness and this is an issue for our enrichment application. In other words, we did not find the topics detected by the existing approaches to be semantically consistent enough for our problem. Hence, we are interested in proposing an approach of topic detection which targets the notion of semantic relatedness between keywords and consequently returns semantically consistent topics. As in the keyword extraction approach, our topic detection approach aims to be domain-independent and easily tunable to different languages.

1.4 Outline

This thesis is organized as the following:

Considering the two main components of our enrichment approach, *i.e.* keyword extraction and topic detection, we present the related works in these two domains. In Chapter 2, we first present various applications of keyword extraction and the different types of lexical units and semantic properties that are usually targeted. We then introduce the traditional evaluation measures and benchmarks in the domain. A wide range of the extraction approaches are presented by distinguishing the extraction features, which assess the “keyness" of lexical units, and the extraction methods that exploit these features.

In Chapter 3, we study the main categories of topic detection approaches along with their properties. The main evaluation measures and benchmarks in the domain are also briefly presented. More specifically, we focus on term similarity measures, which help to capture the relation between pairs of words or phrases.

The overall methodology is introduced in Chapter 4 and all the steps of our proposed approach are presented briefly. The two main components of the approach are then ex-

plained in more details in the following chapters. In Chapter 5, we introduce our keyword extraction approach. Chapter 6 presents the evaluation of this approach and also compares its performance with respect to a baseline approach. In Chapters 7 and 8, respectively, we explain the topic detection approach proposed in this thesis and show the experimental results, evaluations and comparisons with a baseline approach.

Finally, we conclude our work in Chapter 9 by summarizing the main contributions, discussing the main findings and presenting the steps to improve the approach in a future work.

Part II

State of the art

Chapter 2

State-of-the-art in keyword extraction

Contents

2.1 Introduction

Keywords, which are important phrases within documents (Turney, 2000), play an important role in different applications of Text Mining, Information Retrieval (IR) and Natural Language Processing (NLP). With the growth in the quantity of available documents, it is no longer possible for a user to read them all in details. Hence, knowing about the subject of the documents without analyzing them in depth is essential and having an automatic approach to keyword extraction is a necessity.

The task of keyword extraction can be simply defined: given one document or a set of documents, what are the lexical units that best represent them? However, automatically extracting keywords is challenging due to the complexities of natural language, heterogeneity in the type of input documents and diversity of the target applications. After years of active research and development, numerous methods and tools have been designed. They target different applications ranging from lexical resource design to translation, text summarization, and metadata enrichment and they have been tested on various kinds of input data, including long scientific articles, abstracts, web pages, etc.

However, no approach really emerges as the dominant or standard one and it is difficult for newcomers to select the approach that fits the best their problem, input data, and application.

In this chapter, we give a general and comprehensive introduction to the abounding field of term and keyword or keyphrase extraction[1]. This introduction is not bound to any specific application or type of document nor advocates in favor of any specific approach. Considering the general issue of extracting key elements from unstructured documents with content expressed in natural language, we present various solutions that have been proposed over the years. This review of existing approaches helped us to design a method adapted to our specific enrichment problem (see Chapter 5).

2.2 Applications

Many Natural Language Processing (NLP) applications require to extract "key" words and phrases from unstructured textual data. It is interesting to understand how keywords can be exploited once they have been extracted and which types of keywords are targeted.

Design of lexical resources Lexical resources, such as domain specific dictionaries, terminologies or term bases[2], are traditionally used to manage technical documentation (lexical recommendation, technical writing, indexing) and to help expert interaction in scientific and technical domains (*e.g.* aeronautic). Besides more conceptual approaches, textual terminology popularized corpus analysis as a way to design terminological resources and to catch up with terminological evolution, giving rise to the new field of computational terminology in the 90s and to various extraction methods (Bourigault et al., 2001).

Translation It is often for translation and localization that terminological databases were initially created. Multi-word pairs are essential elements in multilingual resources to catch the idioms of specialized languages and to overcome word-to-word translation.

[1] This chapter is based on a paper written by Nazanin Firoozeh, Adeline Nazarenko and Fabrice Alizon that has been submitted to "Natural Language Engineering".

[2] *E.g.* AGROVOC multilingual agricultural thesaurus `http://aims.fao.org/vest-registry/vocabularies/agrovoc-multilingual-agricultural-thesaurus`

Since phrase-based approaches perform better than word-based ones in statistical machine translation, many approaches have been proposed to extract pairs of phrases from aligned or comparable corpora (Déjean et al., 2002, Lefever et al., 2009).

Document summarization Automatic summarization is often based on extraction: the key elements (words, phrases, sentences) extracted from a source document are assembled to form a target document much shorter than the source one. Keyword and key phrase extraction tools are core elements in extraction-based summarization (Wan et al., 2007), key sentences being those that contain the more and the most significant key words and phrases.

Metadata enrichment Keywords are often used as metadata to enrich documents. They can be directly extracted from the source document to emphasize its most important elements or derived from a larger corpus to bring in contextual information. The metadata gives an explicit and computer-processable description of the text content that plays a central role in content management tasks (browsing, indexing, topic detection, classification) and semantic-aware applications (exploratory search, recommendation, reputation analysis or contextual advertising (Mori et al., 2004).

Depending on the target application, keywords of different lengths are targeted. Short keywords, with one or two tokens, are often preferred in lexical resources, whereas Frantzi et al. (2000) consider only compound keywords for indexing digital libraries. Translation, metadata enrichment and summarization need multi-token keywords as well, in order to cover specific or more informative concepts. Keyword length also seems to vary from one domain to the other: Medelyan (2009) reports that agricultural terms are significantly shorter than medical, physics, and computer science ones. A wide variety of length can be observed in the literature on keyword extraction: single token (Liu et al., 2009), two tokens (Muñoz, 1997), up to three tokens (Frank et al., 1999, Hulth, 2003), four tokens (Matsuo and Ishizuka, 2004) or even much longer (Hussey et al., 2011). For our enrichment application, we are interested in extracting both single and multi-token keywords in order to target generic and specific information within sources of information. As will be explained in Chapter 5, our approach extracts keywords with up to five tokens.

In this chapter, we also point out the diversity of the textual sources exploited for extraction: monolingual or multilingual sources, single documents or heterogeneous corpora, long *vs.* short documents, traditional well-written documents or informal ones, mono- or multi-authored documents, scientific, technical, legal or commercial sources. Some authors focus on a specific type of documents such as technical or academic papers or journals (Hussey et al., 2011, Ohsawa et al., 1998a), abstracts of academic papers which are much shorter (Chang and Wu, 2008, Hulth, 2003) or web pages (Kelleher and Luz, 2005, Yih et al., 2006). Some others, however, target several types of documents: Turney (2000) extracts keywords from articles, email addresses and web pages; Renz et al. (2003) approach is applicable on customer feedback statements, intranet documents and news articles.

This chapter focuses on keyword extraction approaches, without targeting any specific application. We put no restriction on the length of the keywords, on the type of documents that are processed or on their language. We consider any type and size of source text, taking into consideration methods designed for processing academic papers as well as web pages. We consider "unstructured data", *i.e.* documents which content is mainly expressed in plain natural language, regardless of their structuration into paragraphs, sections or chapters. In spite of this presentation, our approach has been proposed for extracting keywords from web pages, since the goal of this thesis is to enrich the content of web pages.

2.3 From terms to keywords and key phrases

One major source of complexity in the domain of keyword extraction is related to the diversity of the target elements. Actually, the extraction methods aim at extracting "key elements", which refer to "important" textual units but there are various ways to assess the importance of those elements and various types of units can be targeted, from words to phrases and to sequence of words to sentences. As all the methods are not equally suited to all types of elements, it is important to specify the target elements to determine how to extract them.

2.3.1 Definitions

Terms The notion of "term" refers to the field of terminology where a term is a lexical unit – word or compound – symbolizing a concept (Sager, 1990). Terminology aims at analyzing the concepts and conceptual structures used in a given domain of activity and at compiling the terms denoting those concepts. In computer science and information retrieval, "term" is related to documents rather than domains. It is often used as a synonym of "index term" or "descriptor": terms are expected to describe a document content and are part of a controlled or indexing vocabulary.

Keywords and key phrases The terms "keyword" and "key phrase" do not refer to any theory. An element is considered as a "key" element with respect to a document, when it is an important descriptor of the document content. The opposition word *vs.* phrase simply refers to the mono or multi lexical structure of the textual units, which can be composed of one or several tokens. However, the formal word/phrase distinction is often blurred.

Named entities Named entities are often confused with terms and keywords. The term "named entity" has been coined in NLP during the 90s. It refers to terms that are used as "rigid designators" (Kripke, 1972) or proper nouns which stand autonomously for their referent. Strictly speaking, they are "names of entities" but the term "named entity" is widely used.

In this thesis, the term "keyword" does not refer to any linguistic or semantic theory. "Keyword" stands for any key textual unit that can be composed of one or more words and may work as a common or proper name.

In most cases, the difference between terms and named entities is blurred, the extracted key elements pertaining to both categories. For that reason, in the following, we do not focus specifically on term extraction nor on named entity recognition, although there has been a long tradition of work on the former topic (Jacquemin and Bourigault, 2003) and much effort put on the later (Nadeau and Sekine, 2007). The extraction methods presented in Section 2.6 generally apply to both categories of phrases.

We consider domain and language dependent as well as independent methods. Whenever the applicability and the performances of the reviewed methods depend on these criteria, we try to make it explicit.

2.3.2 Keyness properties

In keyword extraction, the goal is always to extract important or "key" elements, but "keyness" is an elusive concept which interpretation depends on the target application. It can be associated with various properties, even if they are usually not equally important in all contexts.

Univocity is an important feature (Pearson, 1998). Terms are expected to be less ambiguous than common words (*e.g. aircraft vs. plane*), since they are words or phrases whose semantics is stable within a given community and/or context of use. For instance, in aeronautics, *(flight) recorder* always refers to the same type of electronic devices; any domain expert knows what the term means and prefers it to *black box*, which is more colloquial but more ambiguous.

This semantic stability may be contextual but, in a given domain, a term is expected to refer to a single and well defined concept. This semantic stability is also an important property of proper nouns or named entities, as they are often referred to in NLP (*e.g. Airbus A380*). Although the semantic behavior of common terms and proper nouns is different[3], both types of units are good descriptors, due to this stability property.

Representativity In keyword extraction, the importance of the elements is usually assessed with respect to the document source from which they are extracted. Extraction tools aim at identifying the units that have a high "semantic weight" in the source text. Those units are expected to be good descriptors and to properly reflect the informational content of the texts. Keyword extractors usually have to find the right trade-off between the size of the list of extracted units, which must often be minimized, and the representativity of the document description that it gives.

[3] A term denotes a concept whereas a proper noun refers to a specific referential entity (a specific aircraft model).

Well-formedness is not equally important for all use cases. Whenever, the extracted elements are presented to humans, it is usually considered as essential that they are well-formed words or phrases (*e.g. legal right*, *emphasis* or *emphasize* rather than *legal right to* or *emphasi*). However, truncated keywords are very useful in information retrieval and statistical machine translation exploits aligned pairs of word sequences that often do not correspond to any known linguistic pattern.

Cohesiveness is an important property for multi-token extracted units. One does not want to extract any sequence of words, but only those which correspond to semantic units. For instance, in legal language, *trial jury* or *beyond a reasonable doubt* do not have the same degree of cohesiveness than *invest ... with the power or legal right to ...*. The strength of the word association is higher for *trial* and *jury* or for *beyond* and *doubt* than between *invest* and *right*. The components of the latter can be used in association with many different words, whereas the former terms are more idiomatic and less flexible. Cohesiveness goes along with a certain degree of rigidity. In strongly cohesive phrases, one word can hardly be substituted by another[4] and it is often difficult to introduce adjunct elements such as adjectives or adverbs[5]. Correlatively, cohesive phrases are always translated in the same way, even if not on a word-to-word basis (*mock exam / examen blanc*).

Specificity Terms, words and phrases are also considered as key elements of a document or a domain by contrast with other documents or domains. To represent the content of a document, it is important to select only those terms and phrases that are really characteristic of that document and to let aside common textual units that one can find in any similar document. For instance in legal documents, *common law* or *disclosure statement* are rather common terms and give little information on the content of the documents beyond categorizing them as legal ones. Similarly, *user guide* might be considered as a cross-domain, rather than as domain specific term.

Lexical centrality refers to the position of a keyword within its semantic network, the underlying intuition being that the more connected the keywords are to other ones, the more central they are. This property is important to take into account when one expects keywords to summarize a document or a domain. A central unit that is semantically connected to many other ones better summarizes a document or lexical filed than an isolated one. Moreover, if only a limited number of keywords are allowed, one usually tries to maximize the semantic coverage and avoid redundancy at the same time. This can be achieved by splitting the semantic network into subgraphs or "topics" and by selecting only the most central keyword(s) of each topic.

The above semantic properties do not have the same relative importance for all applications. If one aims at designing a terminology or a thesaurus, focus should be put on

[4] One can found *beyond doubt*, *beyond a doubt* or *beyond a reasonable doubt*, but a *doubt* can only be qualified as *reasonable*) and standard variations like singular/plural ones are unusual (*beyond doubts* is by far less frequent than *beyond doubt*).

[5] *E.g.*, a *doubt* may be *reasonable* but not *really reasonable*.

well-formed terms – possibly complex ones – that are important for a domain (representativity for the domain). Univocity and cohesion are also *a priori* more important than specificity and lexical centrality. On the contrary, if one wants to enrich a document with few extracted descriptors, one should focus on representativity to the source document and specificity. In addition to these two properties, our keyword extraction approach focuses on the cohesiveness property, since the structure of the extracted multi-token keywords must be taken into consideration as an important factor. We note that centrality is also an important property in an enrichment task. We take this property into account in the topic detection step of our enrichment approach, where we generate a graph of the keywords extracted by the keyword extraction approach.

Termhood or keyness is a semantic notion, that is difficult to capture. The above properties can only be approximated though surface or formal features (Section 2.5). This explains the variety of keyword extraction methods that have been proposed (Section 2.6) but also the complexity of evaluation issues (Section 2.4).

2.4 Evaluation of keyword extraction

Before entering the description and comparison of approaches that have been proposed, we present the methods, measures and key benchmarks that are commonly used for evaluating keyword extraction approaches and measuring progress. Unsurprisingly, the lack of homogeneity of evaluation methodologies reflects the diversity of the target applications and extraction goals.

2.4.1 Evaluation methods

When the extracted keywords are directly used in another module, their quality can be measured indirectly through their impact on the performance of that module. The evaluation of keyword extraction, however, usually relies on human judgements. Most often, experts are asked to label *a posteriori* the extracted units as "keyword" or "non-keyword". However, *a priori* judgements can also be used: when keywords have been assigned to documents, evaluation consists in measuring the match between the expert-assigned and the extracted keywords[6].

Human judgments must be used carefully, however. Experts are often asked to depart keywords from non keywords but keyness or termhoodness are not a binary notions and human judgments include an element of arbitrariness and subjectivity. It is therefore important to involve different human judges in evaluation and to calculate the degree of agreement among them. Kappa's statistics (Viera and Garrett, 2005) and Fleiss' kappa statistics (Fleiss et al., 1971) are widely used for measuring the inter-evaluator agreement.

[6]Note that, generally, the match with the automatic extraction methods cannot be perfect, as expert assigned keywords are not necessarily extracted from documents.

2.4.2 Evaluation measures

In the state of the art on keyword extraction, different measures and protocols are used for evaluating the extracted keywords. The most traditional measures are *Precision*, *Recall* and *F-measure*. *Precision* is the frequency with which retrieved keywords are relevant (Equation 2.1) and *Recall* is the frequency with which relevant keywords are retrieved by the evaluated approach (Equation 2.2). To have a trade-off between precision and recall, *F-measure* is calculated (Equation 2.3)[7].

$$Precision = \frac{Retrieved \cap Relevant}{Retrieved} \tag{2.1}$$

$$Recall = \frac{Retrieved \cap Relevant}{Relevant} \tag{2.2}$$

$$F_\beta = (1 + \beta^2).\frac{precision.recall}{(\beta^2.precision) + recall} \tag{2.3}$$

Hasan and Ng (2014), however, consider the improvement of the evaluation scheme as a remaining challenge of the field. As a matter of fact, the above traditional measures are questionable. They presume the existence of a "gold" standard but one often has to deal with a long lists of less or more relevant keywords which cannot be considered either as a "gold" reference or as a standard. It is also important to take into account the ranking of the extracted keywords.

In order to overcome those limitations, other evaluation measures have been proposed. Zargayouna and Nazarenko (2010) designed an evaluation protocol that allows for tuning the results to the reference keyword list in order to overcome the rigidity and arbitrariness of existing gold standards. In case of having a ranked list of keywords, *Precision@K* measure can be used. This measure ignores keywords ranked lower than K and computes the precision value over the top-K keywords of the list. Singhal et al. (2017) use the precision@K measure in order to compare the different approaches to keyword extraction for different values of K.

In (Witten et al., 1999), the quality of keywords is assessed by counting the number of matches between the extracted and the author-assigned keywords, rather than by precision and recall. They argue that the simple count is easier to interpret, less misleading and also applicable to the cases that a fixed number of keywords are returned.

Chen et al. (2009) defined two application-dependent measures, named *success rate* and *top-one rate*, to find the shortest keyword which brings the studied web page to the top of the search engine result.

In Chapter 6, we will present our evaluation protocol, where *a posteriori* evaluation is performed by an evaluator and accordingly precision value is measured. We compute the

[7] *F1-measure* is the traditional and the most widely used *F-measure*, which calculates the harmonic mean of precision and recall. Depending on the target application and its sensitivity to afford wrong instances or to miss correct ones, β takes different values in calculation of *F-measure*.

recall value on a gold standard set that we generate using different approaches and tools. Although this gold standard set may not contain all keywords of documents, we believe that it can be used for computing an approximate recall value.

2.4.3 Benchmarks

The cost of collecting training or evaluation data motivates the publication of datasets as public benchmarks, which can be used for comparing approaches or confronting them to state-of-the-art methods.

For instance, SemEval – an ongoing series of evaluations designed for computational semantic analysis systems – proposes in 2010 and 2017 tracks[8] for keyword extraction. A specific experimental setting is defined. At the end, the participating systems are ranked based on their performance.

There are two categories of benchmarks for keyword extraction, depending on whether the keywords assigned to the documents are freely chosen or belong to a controlled vocabulary. Tables 2.1 and 2.2, respectively, give examples of widely used public benchmarks in each category. Among the presented benchmarks, the one developed by Hulth (2003) contains both uncontrolled and controlled keywords, where any suitable keyword and thesaurus unit can be assigned to the documents. The other vocabulary-based benchmarks presented in Table 2.2 and the CiteULike-180 benchmark in Table 2.1 have been developed by Medelyan (2009).

One should choose a benchmark carefully. The type of documents that compose the benchmark and their length are important features to take into account, as extractors are often designed for a specific type and length of documents. Actually, some benchmarks like that presented by Hulth (2003) are made of abstracts of scientific papers, while others are composed of longer documents or web pages.

There are also important features related to how keywords are assigned to documents: the number of annotators involved in the selection of the keywords, their degree of expertise, the guidelines they have to follow regarding the number of keywords to extract and their length. They have an impact on the reliability and homogeneity of the resulting keyword lists.

Length of the extracted keywords mainly depends on the domain under study. According to Medelyan (2009), physics and medical terms are mostly longer than agricultural ones. This property, *i.e.* the length of keywords, is less affected by the type of annotators, *i.e.* readers or writers of the target documents. As reported by Wan and Xiao (2008b), the average length of the keywords assigned by readers is 2.09. Caragea et al. (2014) have also reported that almost half of their author-assigned keywords are bi-grams and they rarely appear as tri-grams or longer n-grams. In general, making use of both author and

[8] The task 5 of SemEval-2010 and the task 10 of SemEval-2017 are respectively presented by Kim et al. (2010) and by Augenstein et al. (2017).

[9] Training set

Table 2.1: Examples of public free-text benchmarks

Title/Generator *Annotators*	Type of docs	Docs number – length	#Tag/Doc
Hulth (2003) *Professional indexer*	Abstracts from Inspec database	2000 – 115 words (avg)	9.63
Nguyen and Kan (2007) *Authors & readers from school of computing*	Scientific papers	211 – 4-12 pages	10
Wan and Xiao (2008b) *graduated students*	News articles from DUC2001	308 – 740 words (avg)	10 (max) 8.08 (avg)
CiteULike-180 *Readers*	Publications from CiteULike website	180 – n/a	5 (avg)
SemEval-2010 *Authors & 50 students*	ACM conference & workshop papers	284 – 6-8 pages	15
Caragea et al. (2014) *Authors*	WWW & KDD titles & abstracts	790 – n/a	4.87 (WWW) 4.03 (KDD)
Augenstein et al. (2017) *undergraduate students*	ScienceDirect publications abstracts	350^9 – 185 words (avg)	22.6 (avg)
Sterckx et al. (2017) *annotators of various ages and backgrounds*	Online Sport & news Lifestyle Magazines Printed press	6908 – n/a	13.8 (avg)

reader assigned keywords can be useful for evaluating an approach. Caragea et al. (2014) point out that in spite of the expertise of the authors, in some cases, they may over express important keywords in different ways. Therefore, it is recommended to use more than one way of annotating the gold standard set. As an example, Kim et al. (2010) make use of both author and reader assigned keywords in generating the SemEval-2010 dataset. Nevertheless, they found a degree of overlap between the two sets of keywords.

In some benchmarks, the assigned keywords are found in the documents. Some other benchmarks, however, may contain keywords out of vocabulary of the studied documents and so the recall value of different approaches to keyword extraction can never reach 100% on these benchmarks. In this case, people mostly report the *reachable recall* value,

Table 2.2: Examples of public vocabulary-based benchmarks

Title/Generator *Vocabulary/Thesaurus*	Type of docs	Docs number – #words	#Tags/Doc
Hulth (2003) *Inspec*	Abstracts from Inspec database	2000 – 115 words (avg)	4.47
NLM-500 (2009) *MeSH*	Biomedical research articles	500 – 4,500 (avg)	15 (avg)
FAO-780 (2009) *Agrovoc*	Documents from FAO's repository	780 – 30,800 (avg)	8 (avg)
CERN-290 (2009) *HEP*	Physics docs from CERN server	290 – 6,300 (avg)	7 (avg)
WIKI-20 (2009) *Wikipedia titles*	Computer Science papers	20 – n/a	5 (min)

computed over the keywords which are found in the studied documents. As an example, in the dataset developed by Hulth (2003), both controlled and uncontrolled keywords may or may not be in the studied abstracts. In SemEval-2010, the readers were asked to assign keywords only from the content of the documents. Analyzing the test set keywords, consisting of 100 documents, showed that 15% of the assigned keywords were not found in the texts. This value was, however, less than the one for author-assigned keywords (19%).

In real applications, one may aim to evaluate robustness of an approach on different types of documents within different domains. In this case, multiple benchmarks should be used to meet the requirements. New benchmarks should also be generated in order to cover more applications, properties and languages. Actually, the existing ones are a bit all the same: most of them are in English, which is a problem in evaluating approaches on non-English documents; most of the public ones contain scientific papers and cannot be used for evaluating keywords extracted from web pages; most of the benchmarks also contain a single type of documents all from the same domain. The large corpus proposed by Sterckx et al. (2017) is a noticeable exception as it combines different types of document targeted for a "diverse and layman audience", but it has nevertheless been designed for a specific application (metadata enrichment).

Our extraction approach has been proposed for extracting both generic and specific keywords from web pages and we aim at evaluating it specifically on French language. To study the robustness of the approach over different domains, we need to perform it on websites with various domains. However, we did not find any public benchmark which satisfies all these properties, *i.e.* containing multi-domain French web pages with both generic and specific keywords assigned to them.

Among different non-English benchmarks, we can refer to the ones provided by Défi fouille de texte (DEFT), which is a French scientific evaluation campaign. More specifically, DEFT2012[10] and DEFT2016[11] focus on the task of evaluating keywords, extracted by different participant systems. In 2012, the training and the test corpus consist of 234 full scientific papers published in journals of Humanities. To evaluate the different methods, their extracted keywords were compared with the author-assigned keywords. In 2016, the keywords extracted by participant systems were compared with the ones assigned by professional indexers. Unlike DEFT2012, the corpus of DEFT2016 challenge consists of four different domains: linguistics, information science, archeology and chemistry. However, as in DEFT2012, the documents of the corpus are all from the same type, i.e. scientific papers[12]. Due to the limitation on the type of documents, these benchmarks are not ideal for our purpose. Nevertheless, since DEFT2016 benchmark contains multi-domain documents, as a future work, it would be interesting to evaluate the effectiveness of our proposed approach on it.

[10] https://deft.limsi.fr/2012/

[11] http://deft2016.univ-nantes.fr/accueil/

[12] In DEFT2012, full papers are analyzed, while in DEFT2016 only titles and abstracts are taken into consideration.

Table 2.3: Morpho-syntactic feature values associated to the word "Cities". NNS represents a plural noun based on Penn Treebank tag list.

Feature	Value	Feature	Value
Token	Cities	Part of Speech	NNS
Lemma	City	Number	Plural

Due to the mentioned limitations, we generate our own experimental data for evaluating the proposed keyword extraction approach. This data is explained in more details in Chapter 6. .

2.5 Extraction features

As the above semantic properties are difficult to exploit as such in keyword extraction, they are approximated through a variety of features that can be derived from a formal analysis of the source text. Depending on the type of input data and the target application, keyword extraction approaches make use of different types of features. This section presents the main ones. We also briefly mention the features exploited in our keyword extraction approach. Details of these features will be presented in Section 5.2.

2.5.1 Morpho-syntactic features

Extraction tools first exploited morphological and syntactical features of textual units. All types of words and word sequences do not have the same probability to be selected as keywords. For instance, some parts of speech, such as nouns and verbs, are more likely to appear in keywords than others, like adverbs and determiners, as they provide more information about the text under study.

Extraction methods rely on plain words (tokens) but also on their lemmatized forms, their parts of speech (POS tag) and some of their morphological features, such as gender or number (singular, plural). See Table 2.3 for an example. The syntactic structure of word sequences is also important to take into account, as the sequences which do not correspond to well-formed syntactic phrases are usually discarded from the keyword candidate lists.

Even syntactic relations (or dependencies) are exploited in keyword extraction: besides keyword cohesiveness, they give an indication of how keywords are related to each other: for instance, *relational database* and *XML database* are co-hyponyms of *database* and *software companies* employ *software developers*. This relational information is especially useful for measuring the semantic relatedness and lexical centrality of keywords. It is also important for the design of structured semantic resources.

Although morphological and POS features are language-dependent features, they are available for a large family of languages for which reliable morpho-syntactic taggers exist. Advanced syntactic features, which require the parsing of the source text, are less widely available as the performance of parsers varies greatly from one language to another. The

performances of those tools often degrade on texts which contain lots of technical or out-of-vocabulary tokens or on non-standard language. In the absence of reliable parsers, extractors often rely on a shallow analysis and simply check that the extracted phrases match any known sequence of morpho-syntactic categories (morpho-syntactic pattern).

In this work, we make use of the morpho-syntactic feature in order to filter out uninformative words and consequently to extract more descriptive keywords. Although this feature is language-dependent, due to the existence of a wide range of taggers for various languages, our approach is easily tunable to new languages.

2.5.2 Statistical features

Statistical features were introduced in the beginning of the 90s (Smadja, 1993) to overcome the limitations of morpho-syntactic methods and to approximate various keyword semantic properties. They are widely used on large corpora, even if rarely in isolation: they are mainly language and domain independent and most of them are easy to compute, even on big data.

Frequency-based features

Term Frequency (TF) (Luhn, 1957) is a very low-level but common statistical feature. The assumption behind its use in keyword extraction task is that the more important a term is in a text (representativity), the more frequent it appears in it. Of course, this feature is both more reliable and more useful when processing long documents than short ones.

Since term frequency strongly correlates with the size of documents, one usually considers a *Normalized Frequency* (NF). Equations 2.4 and 2.5 show two traditional formulae of normalized frequency for a given text. Another limitation of TF feature is that it does not depart grammatical or common words from content ones as the former are usually highly frequent.

$$NF_{w_i} = \frac{\#\,of\ occ.\ of\ w_i}{Total\ \#\,of\ word\ occ.} \tag{2.4}$$

$$NF_{w_i} = \frac{\#\,of\ occ.\ of\ w_i}{\#\,of\ occ.\ of\ the\ most\ frequent\ word} \tag{2.5}$$

Normalized frequency is one of the statistical features exploited in our keyword extraction approach. In general, the average length of web pages is not very short and so the term frequency can bring information about their constituent words. Moreover, the search engine optimization techniques (SEOmoz, 2012) recommend people to put important words in different sections of web pages and so to use them frequently. Hence, term frequency is an effective feature for representing the importance of words in web pages. To not be affected by very generic and common words, we define a list of such words and filter them out in the extraction procedure to eventually extract more specific keywords.

Inverse Document Frequency (IDF) (Sparck Jones, 1972) is a quantity borrowed from Information Retrieval. It is defined by Equation 2.6, where the document frequency (DF) of a term corresponds to the number of documents in a target collection in which it occurs. The IDF is lower for the common terms that appear in many documents of the collection and higher for those which have a low document frequency. This measure provides a valuable indication of the specificity of a term in relation to a document but using IDF requires a collection to which the document can be confronted.

$$IDF_{w_i} = log(\frac{1}{DF_{w_i}}) = log(\frac{Number\ of\ documents\ in\ the\ collection}{Number\ of\ documents\ in\ which\ w_i\ occurs}) \tag{2.6}$$

Term and document frequencies are traditionally combined in the *Term Frequency-Inverse Document Frequency* feature (TF.IDF, Equation 2.7), which is widely used in Information Retrieval (Salton et al., 1975). The IDF factor tends to counterbalance the high TF value of common terms present in most documents and higher the weight of words that appear rarely in the rest of the collection.

$$TF.IDF_{w_i} = TF_{w_i} * IDF_{w_i} \tag{2.7}$$

Specificity of keywords within a target domain can be also captured through *Domain Relevance* (DR) score proposed by Navigli and Velardi (2002). The domain Relevance of a term is high if it appears frequently in the target domain and rarely in other domains. The formula for computing the domain relevance of term t is presented in Equation 2.8, where D_i is the target domain, containing a set of documents, and $D_1, ..., D_{i-1}, D_{i+1}, ..., D_N$ represent other domains.

$$DR_{D_i}(t) = \frac{freq(t, D_i)}{\max_j(freq(t, D_j))} \tag{2.8}$$

Although the specificity measures, notably TF.IDF, have been widely used, we do not exploit them in our approach. The main reason is that in our enrichment approach, a corpus of documents, from the same or different domains, may not be always available and so we need to extract keywords of a document without using any corpus.

N-Gram-based features

N-Grams are sequences of elements extracted from a text. They can be defined either at the character level or at the word level.

Character-level N-Grams are seldom used in keyword extraction but they help to identify recurring words in spite of small variations, like the plural/singular alternation or words belonging to the same stem (*e.g. visualize, visualizing, visualization*) (Cohen, 1995). This feature helps to identify morphological word classes when no lemmatizer or stemmer is available. It is mostly used for single-token extraction, except for languages like German in which the single/multi-token distinction is blurred due to the frequency of compounds.

The motivation behind using *word-level N-Grams* is to study all possible sequences of words within a text and select the important ones as candidate keywords. This allows to skip POS tagging and parsing but it generates many candidate keywords, some of which with invalid grammatical patterns. For most applications, the resulting list must then be filtered out, a filtering step which adds complexity to the extraction process. Due to the complexity of using *N-grams*, we rather use POS taggers as we found them available and effective for our studied languages.

Sequential frequent patterns[13] can be considered as a variant of N-Grams but they can be of arbitrary length (whereas N-Grams are limited to N elements). They can also be discontinuous, which is useful to abstract from the surface variations of terms. However, this approach raises the same overgeneration problem as the N-Gram one and a much higher computation complexity.

Co-occurrence-based features

Term or word co-occurrence is a statistical feature designed to catch word associations or keyword cohesiveness. The basic idea behind using this feature is to capture words which tend to appear together within a given type of context. The basic formula is given by the following equation[14], where c is a type of context:

$$Cooc_c(w_i, w_j) = \#\ of\ contexts\ c\ in\ which\ w_i\ and\ w_j\ co-occur \tag{2.9}$$

Co-occurrence features mainly differ by the type of context which is considered for the co-occurrences, *i.e.* the context in which two terms are considered as co-occurrent. The computing complexity of the extraction process increases with the size of the context. Momtazi et al. (2010) list four categories of term co-occurrence features that we present bellow.

In *Sentence-wise co-occurrence*, only terms occurring in the same sentence are said to be co-occurrent (c = sentence) but it is often less costly to take a smaller and a fixed context into account. In *Window-wise co-occurrence*, a sliding window with a fixed size is set as an input parameter (c = window of n words). Any time two terms appear in the same window, one co-occurrence is counted. Of course, it is also possible to restrict to well-formed syntactical phrases to avoid accidental co-occurrences of unrelated terms (c = well formed phrase) but *Syntax-wise co-occurrence* requires the initial chunking or parsing of the text. When the context gets much larger and the whole document is taken into account, the extracted pairs of words are not as strongly associated as in the previous cases. *Document-wise co-occurrence* gives weak semantic association of words instead of cohesive keywords. The feature is nonetheless useful as an indication of the semantic similarity of words.

[13] Cellier et al. (2014) give an interesting overview of these works.

[14] Co-occurrence can be defined as a binary feature (presence/absence of co-occurrence) rather than as a scalar one (number of co-occurrences). The actual formulae are usually more complex: they are sensitive to word order ($Cooc_c(w_i, w_j) \neq Cooc_c(w_j, w_i)$) and only significant co-occurrence scores are considered.

As will be explained in Chapter 5, we make use of the window-wise co-occurrence feature in order to capture the association between single words and to generate multi-token keywords. Since some sentences of web pages are long, we did not find the sentence-wise co-occurrence effective enough for capturing this association.

Park et al. (2002) proposed *Lexical Cohesion* (LC), which computes the association between multi-word terms. This measure generalizes the Dice coefficient and is computed using Equation 2.10. In this equation, $|T|$ is the total number of words in term T and w_i is its constituent word.

$$LC_T = \frac{|T| \times freq(T) \times log_{10} freq(T)}{\sum_{w_i \in T} freq(w_i)} \tag{2.10}$$

The lexical cohesion of a term in a text is high if its constituent words appears more often within the term than individually.

Other proposals have been made to measure word similarity. They are often based on the comparison of word vectors, which show the distributions of terms within a corpus (Harris, Hindle, 1990). Cosine similarity, Dice coefficient and Jaccard index (Manning and Schütze, 1999) are examples of these measures. Considering A and B as two word vectors, Equations 2.11, 2.12 and 2.13 respectively show the formulae for computing these measures.

$$Cosine(A, B) = \frac{A.B}{||A||_2||B||_2} \tag{2.11}$$

$$Dice(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{2.12}$$

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.13}$$

Table 2.4 shows examples of approaches which make use of statistical features. According to the table, frequency-based features are the most widely-used statistical features and can be considered as a "must" feature for developing a new approach to keyword extraction.

Informational features

Informational or textual features are language and domain-independent. They exploit the information clues that authors use to bring attention to important points in their texts. We distinguish four types of informational features, respectively based on typography, document structure, keyword position within the source documents and keyword length.

Examples of *typographical features* are underlined, bold, italicized elements but words and keywords can also be highlighted through quotations marks. Any type of typographical emphasis can be exploited to spot the most relevant keywords. Even if the productivity of

Table 2.4: Examples of approaches exploiting statistical features

Approach	Feature 1	Feature 2	Feature 3
Salton et al. (1975)	TF.IDF		
Cohen (1995)	TF	DF	N-gram
Ohsawa et al. (1998a)	TF	Co-occurrence	
Frank et al. (1999)	TF.IDF		
Frantzi et al. (2000)	TF	DF	
Turney (2000)	TF		
Hulth (2003)	TF	DF	
Matsuo and Ishizuka (2004)	TF	Co-occurrence	
Yih et al. (2006)	TF	DF	
Medelyan and Witten (2006)	TF.IDF	N-gram	
Wan and Xiao (2008a)	TF.IDF	Co-occurrence	
Zhang et al. (2008)	TF.IDF		
Hussey et al. (2011)	TF	N-gram	

these features depends on the type and typographical convention of the source text, they are quite reliable when available.

If the source text is *structured*, one can exploit the fact that some specific text areas are more informative than others. According to the recommendations made for search engine optimization (Google, 2010, SEOmoz, 2012), in web pages, important words are mainly put in sections such as title and Meta description. In our work, we make use of this structural information in order to capture the most important words of a page.

The *position of the keyword occurrences* within the source text – "spatial use of the words" for Herrera and Pury (2008) – can also be analyzed as an indication of the importance of the studied terms. In academic documents, terms that appear in the abstract, at the beginning or at the end of a chapter are expected to be more informative than the others.

Informational features can be modeled as Boolean values (presence/absence of the term in a given area) or as scalar ones (position of the first occurrence of the term or on the average position of the term occurrences), possibly normalized with respect to the length of the text. In any case, informational features are used to measure the representativity of the terms.

The *length* of keywords may also give information about them. In some approaches, keywords with less than three characters are assumed to be uninformative and so are filtered out from the list of candidate terms. An example is the pre-processing performed by Turney (2000), where words with less than three characters are removed. On the opposite, some approaches may assume that longer keywords contain more specific information. Nevertheless, the length property mainly depends on the target application (see Section 2.2). As an example, if abbreviations are important terms in the target application, removing short terms leads to missing important information. Our studies show that the

Table 2.5: Examples of approaches exploiting informational features

Approach	Feature 1	Feature 2	Feature 3	Feature 4
Frank et al. (1999)	positional			
Turney (2000)	positional	typographical[a]	length	
Hulth (2003)	positional			
Yih et al. (2006)	positional	typographical[a]	structural[b]	length
Medelyan and Witten (2006)	positional	length		
Zhang et al. (2008)	positional	structural[c]		

[a] Capitalization
[b] Occurrence in "anchor text", "Meta" tags, "title" tag, "URL"
[c] Occurrence in "title", "abstract", "body (full-text)", "heading", "first paragraph", "last paragraph", "references"

length of keywords is not an effective feature for our problem, since the information that short keywords, such as abbreviations and metrics, provide is valuable in our enrichment application.

Examples of approaches which exploit this informational features are listed in Table 2.5. The "positional" property in the table indicates the position of the first occurrence of the keywords in the studied document. According to the provided examples, this property is the most widely-used among the informational features. Unlike structural property, positional property does not depend on the type of the studied document. Hence, it can be used in a more generic way for targeting various types of documents. Inspired by this finding, we exploit positional feature in our keyword extraction approach.

2.5.3 Resource-based features

The quality of the extracted keywords can also be measured with respect to an external semantic resource, such as dictionaries, thesauri, that provide additional information on the studied words.

Due to the dependency to external resources, these features are considered as domain and language-dependent features.

One can exploit an existing gold-standard terminology to spot the term occurrences in the source text. This amounts to keyword identification rather than extraction but such a *dictionary-based validation* can be exploited to assess the importance or representativity of the terms.

A structured semantic resource also helps to identify *semantic relationships*, such as groups of synonyms or topic-based clusters, which serve for measuring the semantic centrality of the terms, assuming that related terms are more likely to be important than isolated ones. This feature has been used for instance for designing back-of-the-book indexes (Nazarenko and Aït El Mekki, 2007).

Resource-based features are also exploited to assess the specificity of extracted keywords. Drouin (2003) considers that the candidate keywords composed of *domain-specific*

Table 2.6: Examples of approaches exploiting resource-based feature

Approach	Exploited resource
Medelyan and Witten (2006)	Thesaurus
Yih et al. (2006)	Query log
Hussey et al. (2011)	Thesaurus

vocabulary items are more likely to be relevant than others. Ma et al. (2008) exploit query logs as resource and assess the relevance of keywords depending on how frequent they are in queries. Other examples of approaches which exploit this category of features are presented in Table 2.6.

In our work, we aim at performing the enrichment application on various domains. Hence, our method must be knowledge-poor and domain-independent in order to meet this requirement. However, we found out that using the resource-based feature increases the effectiveness of the proposed approach. Since this data is not always available, we use it as an optional feature that is exploited in case of availability. Without this feature, the extraction approach is effective enough.

2.5.4 Conclusion on extraction features

A long list of features has been tested for keyword extraction and only the main categories are presented here. Table 2.7 presents the categories of features used by some of the approaches introduced in this survey, taking all their steps, including the pre-processing, into consideration. As an example, Frank et al. (1999) rely on morpho-syntactic analysis for the pre-processing, but do not use this feature in the core approach.

According to Table 2.7, statistical features can be considered as the elementary features in extraction approaches. Any new approach to keyword extraction exploit these features in order to take advantage of the basic properties of words within the studied document. Among the statistical features, the frequency-based ones are the most exploited.

Morpho-syntactic features are mainly used for reducing the number of candidate words or phrases in the extraction task. Although they are not used as frequently as the statistical features, many works exploit them as a basic feature. On the contrary, resource-based features are not frequently exploited. As a matter of fact, finding a relevant resource can be challenging and for some domains such a resource may not be available. Moreover, domain-independent approaches are often preferred and the resource-based feature does not meet that requirement.

As shown in Table 2.7, the presented features are mainly used in combination and not always on an explicit basis. In fact, most of these features have been introduced on an empirical basis to improve the quality of the extracted keyword lists and it is often afterwards that they have been related to semantic properties and justified.

Table 2.7: Categories of features exploited in example approaches

Approach	Statistical	Morpho-syn	Informational	Resource-b
Salton et al. (1975)	✓			
Cohen (1995)	✓			
Ohsawa et al. (1998a)	✓			
Frank et al. (1999)	✓	✓	✓	
Frantzi et al. (2000)	✓	✓		
Turney (2000)	✓	✓	✓	
Hulth (2003)	✓	✓	✓	
Matsuo and Ishizuka (2004)	✓	✓		
Turney (2003)	✓		✓	
Mihalcea and Tarau (2004)	✓	✓		
Kelleher and Luz (2005)	✓		✓	
Yih et al. (2006)	✓	✓	✓	✓
Medelyan and Witten (2006)	✓		✓	✓
Wan and Xiao (2008a)	✓	✓		
Zhang et al. (2008)	✓	✓	✓	
Hussey et al. (2011)	✓			✓

We make use of these findings in selecting the extraction features for our keyword extraction approach (Section 5.2).

2.6 Extraction methods

This section focuses on the core of keyword extraction, *i.e.* on extraction methods, that take one or several documents as input, possibly exploit external resources and output a (possibly ranked) list of keywords. We do not consider the way the result is used in applications, nor the interaction with the user if any, although not all systems meet the same needs, as mentioned above.

We present five different approaches or categories of methods that have been successively proposed, even if the more recent methods often re-exploit the previous ones.

Independently of that categorization, there are two main strategies for extracting keywords. The synthetic one consists in extracting the relevant keywords at once, regardless of the number of their constituent tokens, and then filters out the least relevant ones.

The analytic approaches, on the contrary, first extract the most relevant single words, which are then extended to surrounding words and/or merged to generate the final list of (possibly long) keywords. Our keyword extraction approach is an analytic approach due to the following reasons:

- We did not find the phrase extractors to effectively extract the lexical units of web pages[15]. This could be due to the fact that while writing the content of web pages, people do not follow the same standard as in other types of documents, such as

[15]We, however, found that taggers achieved rather good results on web pages.

scientific papers. It could be also related to the functionality of some phrase extractors, such as named entity recognition tools, which are case sensitive and cannot be effective on web pages, where words in headings and titles often begin with capital letters.

- We are interested in both common and specific phrases and also in phrases consisting of various parts of speech. However, phrase extractors mainly focus on specific phrases, *e.g.* names of organizations, people, locations, etc., or they extract noun phrases.
- Since in the analytic strategy, keywords are "generated" out of single words, we believe that more combinations of words and so more keywords can be extracted (generated) from a document.

The degree of language or domain dependency is another important factor to take into account: some approaches are domain dependent (Zhang et al., 2008), while others are both language and domain independent (Mihalcea and Tarau, 2004, Salton et al., 1975). In this thesis, we propose a knowledge poor and domain-independent approach, which is easily tunable to different languages.

2.6.1 Basic statistical methods

Some traditional approaches of keyword extraction simply use statistical features in order to extract the most significant terms of a given text. These approaches are mainly language and domain-independent. How naive they may be, they have been widely used and compared to other methods.

TF.IDF is one of the dominant statistical features. (Salton et al., 1975) proposed the Theory of Term Importance and showed that the importance of a textual term depends not only on its frequency (or representativity), but also on its specificity. This feature has often been used for term extraction. For instance, Cohen (1995) applied both TF and IDF features on character-level n-grams extracted from a given document. This approach focuses on single token or sub-token keywords but once the most relevant n-grams are identified they can be extended into words or merged into phrases. TF.IDF has been widely used as a baseline for the evaluation task. We also use this method for this purpose and compare the effectiveness of our approach with respect to it (Chapter 6).

2.6.2 Pattern-based methods

Pattern-based methods generally use morpho-syntactic features and are sometimes enriched with lexical information. These patterns are often expressed as sequences of POS tags and key lexical units. As an example, *Noun+"of"+Noun* is a valid structure in English, while *Noun+Noun+preposition* is not. These methods aim at characterizing the valid linguistic structures of the keywords to extract through surface patterns. They require *a priori*

linguistic knowledge and so are language-dependent. They, however, do not require any training corpus.

In general, the choice of the linguistic filters depends on the trade-off between precision and recall: relaxing a filter lowers the precision but increases the recall. Patterns are usually not used in isolation. They can be used *a priori* to select candidate keywords that are then filtered or ranked on a non-linguistic basis (*e.g.* association measures in (Daille, 1996)). This is a fairly common approach (Dostàl and Jezek, 2011, Frantzi et al., 2000, Hulth, 2003). Patterns are also used *a posteriori* to filter out collocations or extracted/generated keywords that have been pre-selected on statistical grounds but are not syntactically well-formed (Smadja, 1993, Wan and Xiao, 2008a). Our keyword extraction approach applies a pattern-based method *a posteriori* in order to filter out pre-defined structures of keywords. These structures are basically uninformative but frequently generated by the approach.

2.6.3 Supervised methods

Keyword extraction can be regarded as a classification problem, each candidate keyword being labeled as either a keyword or a non-keyword. Supervised methods take training data as input and rely on training features, but they differ in the features used for training classifiers and the types of their classifier(s).

Supervised approach to keyword extraction was first proposed by Turney, who tried two classifiers : 1) C4.5 decision tree (Quinlan, 1993), with twelve statistical and morpho-syntactic features, and 2) GenEx algorithm, itself based on Extractor (Turney, 2000), a keyphrase extraction algorithm that also exploits a combination of twelve statistical and morpho-syntactic features. Results show that GenEx outperforms C4.5. One drawback of the approach is that words with less than three characters are dropped as uninteresting words, which rules out most of the units and abbreviations.

Various improvements have been proposed on Turney's approach. KEA algorithm (Frank et al., 1999) is based on a Naïve Bayes classifier and uses a simpler and a smaller set of statistical features. Turney (2003) improved KEA by increasing the coherence of the extracted keywords using statistical associations between them; using a rule induction approach, Hulth (2003) showed that exploiting morpho-syntactic features improves the performance of the previously proposed supervised machine learning approaches; KEAWeb (Kelleher and Luz, 2005) exploits hyperlink structure to improve the keywords extracted by KEA and KEA++. Medelyan and Witten (2006) uses the semantic information of a domain-specific thesaurus to overcome the limitation due to word synonymy.

Many different machine learning approaches have been tested for extracting keywords: Zhang et al. (2006) applied Support Vector Machine (SVM); Yih et al. (2006) but also Dave and Varma (2010) train classifiers respectively using logistic regression algorithm and a Naïve Base classifier; Sarkar et al. (2010) extract keywords from scientific articles using multi-layer perceptron neural network. According to the results, these approaches

Table 2.8: Examples of the supervised approaches and the categories of their extraction features

Approach	Statistical	Morpho-syn	Informational	Resource-b
Frank et al. (1999)	✓		✓	
Turney (2000)	✓	✓	✓	
Hulth (2003)	✓	✓	✓	
Turney (2003)	✓		✓	
Kelleher and Luz (2005)	✓		✓	
Yih et al. (2006)	✓	✓	✓	✓
Zhang et al. (2008)	✓	✓	✓	

outperform KEA algorithm and Dave and Varma's method Dave and Varma (2010) performs better than that of Yih et al. (2006). Augenstein et al. (2017) confirm these trends.

Zhang et al. (2008) were the first to consider keyword extraction as a string labeling problem and used Conditional Random Fields (CRF) model to extract keywords. Authors showed that CRF model outperforms other machine learning methods such as Support Vector Machine, multiple linear regression model, etc. They exploited statistical and morpho-syntactic features, whereas Chang and Wu (2008) used CRF model with a richer combination of morpho-syntactic, statistic, semantic and informational features.

In Table 2.8, we present some examples of the supervised approaches along with the categories of the extraction features that they exploit. As it is seen, statistical and informational features are both widely used in the supervised approaches. Some approaches also make use of morpho-syntactic features by giving a higher importance to nouns. As mentioned before, using external resources can limit the applicability of the approaches. Consequently, resource-based features are not used very often in the supervised methods.

Supervised machine learning approaches have promising performance in extracting keywords but the training data requirement is a limitation. When no "naturally" annotated data is available, it must be generated manually as for evaluation, and it is not a trivial task. It it not even possible to generate a training set for all types of sources: according to Chen et al. (2009), it is impossible to collect a large enough training data for all types of web content, which prevents using supervised approaches for extracting keywords from web documents. Considering these limitations, we propose an unsupervised approach, which does not require any annotated data and eventually produces more robust results.

In the state of the art, the unsupervised approaches mainly consider keyword extraction as a clustering or a ranking problem. Graph-based and entropic methods are examples of unsupervised approaches.

2.6.4 Graph-based methods

The idea behind graph-based methods is to take into account the connectivity of terms and to capture the semantic centrality of keywords. The overall approach consists in generating a graph of elements and to use it to cluster or rank those elements.

Graph generation

Depending on the type of analysis and the goal of extraction, the generated graphs can be directed or undirected, and weighted or unweighted.

Connectivity can be measured locally, at the document level, or globally on a collection of documents. In the first case, the idea is to exploit the relation and connectivity of the terms within a single document Ohsawa et al. (1998a). However, more distant cross-document relationships can be exploited. Some works make use of both local and global-context to extract keywords of a studied document Wan and Xiao (2008a): the results of Wan and Xiao (2008a) show that adding global context increases the performance of the proposed approach. On the annotated news from DUC2001 dataset, the highest value of F-measure obtained by the best set of parameters was 0.317.

Graph-based approaches also differ in the selection of vertices and edges. Approaches with different goals use different units as vertices of the graph and also various relations and metrics for generating the edges.

Most often, vertices correspond to single or compound keywords of a given document or collection of documents. Keyword candidates are represented as vertices of a graph. The edges reflect the semantic relatedness of the candidates, which is often measured through co-occurrence (two words are connected in a document graph if they co-occur within a certain window in that document). This type of graphs has been popularized by TextRank (Mihalcea and Tarau, 2004) but various algorithms have taken that idea since then. Alternative metrics can be used to capture that semantic relatedness (*e.g.* Jaccard coefficient and the graph can be based on alternative semantic relationships, as in (Grineva et al., 2009), which relies on a semantic relatedness derived from Wikipedia, or that of Huang et al. (2006), which exploits syntactic relations.

A marginal approach considers documents as vertices and relies on the connectivity between documents (Kelleher and Luz, 2005). The goal is to capture the intertextuality expressed through the hyperlinks of web documents, in legal document networks (Mimouni et al., 2015) or in chats (Abilhoa and de Castro, 2014). The underlying idea is that documents related to a document d within a corpus provide additional information for identifying relevant keywords for d.

A more recent approach considers graphs of topics rather than graphs of words or of documents: TopicRank algorithm (Bougouin et al., 2013) has been proposed as an improvement over TextRank. Different benchmarks were used for evaluating TextRank and TopicRank algorithms. The former algorithm was tested over the benchmark developed by Hulth (2003) and the maximum value of F-measure obtained with the best performing method was 36.2%[16]. Bougouin et al. (2013), however, tested their approach on DEFT2016 benchmark. They reported the maximum value of F-measure over four domains of Archeology, Chemistry, Linguistics, and Information science as 40.11%, 18.28%, 24.19%, and 21.45%, respectively.

[16]Hulth (2003) achieved the F-measure of 33.9% on the same dataset.

Graph-based analysis

Different *unsupervised* analyses can be performed on the generated graph to get a list of keywords. They are mainly based on clustering and/or ranking algorithms.

Clustering algorithms are used to cluster the nodes of the graph, each cluster corresponding to a set of variant keywords or a group of semantically related ones. For instance, Ohsawa et al. (1998a) show that clustering a co-occurrence graph outperforms both TF.IDF and N-Gram approaches for indexing. Grineva et al. (2009) apply community detection techniques on a weighted semantic graph to identify topically related terms and to rule out unimportant ones.

The most common approach relies on *ranking algorithms* and aims at ranking the vertices of keyword graph using the global information recursively computed from the entire graph. This approach has been initially proposed for TextRank and was shown to outperform that of Hulth (2003) in terms of precision and F-measure. Mihalcea and Tarau (2004) used PageRank ranking algorithm (Brin and Page, 1998) but claimed that other ones, such as HITS (Kleinberg, 1999), can also be applied. Many variant approaches have been proposed since then (Bougouin et al., 2013, Liu et al., 2010, Wan, 2007). Different benchmarks have been also exploited in different works for the purpose of evaluation. Similar to Mihalcea and Tarau (2004), Liu et al. (2010) made use of the benchmark developed by Hulth (2003) and reported the obtained F-measure value as 0.242 on this benchmark. They also tested their approach on DUC2001 dataset, which resulted in a higher F-measure value over the extracted keywords (0.312).

Structural properties of graphs are also exploited for clustering and/or ranking. The notions of degree, shortest path, centrality, and betweenness are used by Huang et al. (2006) and several authors consider that co-occurrence graphs are similar to small-world graphs (Matsuo et al., 2001). Boudin (2013) compared the efficiency of the PageRank measure with other centrality measures on undirected and weighted word co-occurrence graph. Authors showed that degree centrality performs comparable to the PageRank but that closeness centrality has the best performance on short documents. Lahiri et al. (2014) extended this analysis to directed word collocation and noun phrase collocation networks. Results showed that some other centrality measures, such as degree and strength, perform very similarly, or slightly better, when compared with PageRank and are much less computationally expensive. Authors tested their approach on four datasets, including two of the previously mentioned ones: SemEval2010 and the benchmark developed by Hulth (2003). The highest F-measure values obtained with the best performing centrality measures on these two datasets were respectively 6.32% and 8.97%.

Centrality was also used by Abilhoa and de Castro (2014), who generated undirected and weighted/unweighted graphs from tweet messages in order to extract their keywords.

2.6.5 Entropic methods

Alternative approaches rely on the assumption that "keyness" is reflected in the spatial distribution of the occurrences of words.

Entropic methods are statistical approaches which rely on Shannon's theory of information (Shannon, 1948) to quantify the information content of a keyword in a given text based on the distribution of its occurrences in the text. Based on such an entropy measure, one can rank the keywords of the text and detect the more informative ones. The underlying assumption is that words which are more relevant to the topic of the studied text mostly concentrate in some limited areas to represent author's purpose. On the other side, irrelevant words have random positions throughout the text. The main advantage of this method is that it needs no external information and no *a priori* knowledge about the structure of the studied document. It is also language independent and requires no corpus of documents, as opposed to unsupervised approaches based on TF.IDF.

The first entropy-based approach to automatically extract keywords was designed for literary texts (Herrera and Pury, 2008) and experiments on a large book gave promising results. However, the main challenge of this approach is that the studied text needs to be initially partitioned and, according to Carretero-Campos et al. (2013), the result of extraction depends on the choice of partitions. Mehri and Darooneh (2011) later defined three other entropic metrics.

The underlying intuition of these approaches was validated by Mehri et al. (2015), who compared a studied text with a shuffled text, where words of the studied page were positioned randomly. The authors showed that the spatial distribution of relevant words significantly differs between original text and the shuffled one, whereas for the irrelevant terms, the distributions are very close.

Carretero-Campos et al. (2013) compared the entropic methods to older and simpler "clustering" ones, based on the idea that occurrences of relevant keywords tend to "cluster", whereas basic terms have a more homogeneous distribution, a phenomena which can be easily captured through the standard deviation of the distance between consecutive occurrences of a word (Ortuño et al., 2002) and which does not require any partition of the source text. These clustering approaches seem to perform better on short documents.

2.6.6 Conclusion on extraction methods

Most of the methods proposed for extracting keywords fall into one of the above categories but even in a given category they often differ in the features they rely on and in the type and number of keywords that they aim at extracting. Each method has its own strengths and domain of application but recent works show that combining different types of methods and features leads to more generic and performant keyword extraction approaches. Danesh et al. (2015), for instance, report promising results with an unsupervised method that combines n-grams for candidate generation with various ranking steps respectively based

on traditional statistic features, the position of the first occurrences and a co-occurrence graph. Nevertheless, obtaining a high value of F-measure in the task of keyword extraction remains to be a challenge. According to the results reported in SemEval-2010 (Kim et al., 2010), the state-of-the-art performance in terms of F-measure value is mostly 20% to 30%. Results reported in SemEval-2017 (Augenstein et al., 2017) also confirms the difficulty of the task of keyword extraction.

In this thesis, we aim at extracting keywords from web pages of various domains. Hence, we do not exploit any training data as no single training corpus would be robust to domain diversity. Generating training data for each domain would be also too expensive. As a result, we focus on an unsupervised and knowledge-poor method with a generative process in which keywords are generated out of single words. More specifically, we exploit basic statistical and pattern-based methods. We found this basic method to be robust over different domains and to be executed in a very reasonable time due to its light procedure. The approach is also aimed to be robust to different kinds of noise in the content of web pages. In addition, it must make a trade-off between precision and recall. Nevertheless, since our enrichment application interacts with users, who perform the enrichment, recommending "bad" keywords to users is not acceptable and so precision is a more important factor.

In Chapter 5, we will explain in more details our keyword extraction approach. Details on the obtained precision and recall values are presented in Chapter 6.

Chapter 3

State-of-the-art in topic detection

Contents

3.1 Introduction

Today, the volume of available information is growing rapidly. This information can be provided in different forms, such as web pages on the web, documents in repositories, product reviews in social networks, video and audio data files in digital libraries, etc. Due to the tremendous amount of data and also the growing nature of the available information, each source of information cannot be analyzed manually in depth. Therefore, having automatic approaches for analyzing a large collection of documents is a necessity.

As explained in Chapter 2, one way of dealing with the great number of available textual documents is to extract their "key" words or phrases and to further exploit them as representatives of the documents. Although extracting keywords from documents decreases the complexity of analysis, it does not give an adequate representation of the documents topics. In fact, it is possible that a document discusses various topics. Consequently, keywords extracted from this document belong to different topics and not all these topics might be relevant to be analyzed. In order to detect the different topics within a collection

of keywords and to select the desired ones, the keywords should be topically grouped into clusters for someone to analyze only the relevant clusters of keywords. Detecting the underlying topics in a given collection is referred to as *Topic Detection*. These topics can be used for further analysis and discoveries over the data, including detecting the hot topics within the collection, modeling the structure or the connectivity between the detected topics, and observing the evolution of topics (trends) in case that the data is changing over time. This information can be used in different applications such as document clustering and summarization, question answering, news analysis, market segmentation, sentiment analysis, etc.

Although "topic" is a broad notion that can be defined in different ways, in natural language processing and information retrieval, the definition of a "topic" is basically simplified to a set of items with close semantic relationships. It should be noted that in the literature, depending on the application and the type of the input data, an "item" can be a single word, keyword/keyphrase, document, news story, etc.

Before explaining the main approaches of topic detection in the literature, the difference between topic detection and topic classification should be clarified. *Topic Classification* is the task of grouping documents or keywords into a pre-determined set of topics. Approaches proposed by Lee et al. (2011) and Baykan et al. (2009) are examples of the topic classification task, where the input data has been classified into 18 and 15 pre-defined topics, respectively. Hence, the topics are known to the system *a priori* and the procedure is supervised. Although the number of the pre-determined topics can be only two, topic classification is mainly regarded as a multi-class classification problem, where more than two topics are known to the system. Moreover, the classification task can be single-label or multi-label, where each instance respectively belongs to only one or more than one class (topic). By contrast, *Topic Detection* is an unsupervised task with no prior knowledge about the number and the type of topics. Topic classification suffers from some of the constraints of supervised approaches, *e.g.* predicting the number of topic classes in advance and providing a rich training and testing data for learning and testing the model and making a decision (Tur and De Mori, 2011).

We also emphasize that the goal of our topic detection step is different than that of ephemeral clustering. In ephemeral clustering, also known as search result clustering (SRC), results returned by a search engine as an answer to a query are analyzed and groups of topically relevant web pages are generated. Approaches in ephemeral clustering mainly use web snippets as source of information. They can be text-based (Carpineto and Romano, 2010, Osiński et al., 2004, Zamir and Etzioni, 1998) or knowledge-based (Marco and Navigli, 2013, Scaiella et al., 2012). Although these approaches can be used for categorizing web search results in different applications, such as clustering web images for mobile devices (Moreno, 2015), we do not exploit them for our enrichment problem. The main reason is that our enrichment approach works on the keywords level, meaning that for enriching a document, semantically relevant keywords are required. Ephemeral clustering approaches, however, work on the documents level and documents in the generated clusters may contain

more than one topic. As an example, a web page may contain a description about a product and also payment information. Clearly, these two are not discussing the same domain, but this distinction cannot be detected using ephemeral clustering approaches, which could assign the web page to both "payment" and "product" clusters. Due to this property, keywords extracted from documents of a cluster are not necessarily semantically related and this makes an issue for our enrichment problem. In addition, we aim at performing more analysis on keywords in order to detect finer topics and this is not what ephemeral clustering approaches are expected to do.

One application of ephemeral clustering in our enrichment approach could be to use it as a pre-processing step, mainly having an ambiguous enrichment point of view. More specifically, while generating the enrichment collection according to an input keyword (see 4.2.1), ephemeral clustering could be used to pick the right cluster of documents and to filter out the ones related to other meanings of the keyword. Nevertheless, our topic detection approach does this disambiguation on a collection of extracted keywords. Hence, this pre-processing clustering does not add any advantage to our approach and for complexity reasons, we do not exploit it.

In the following, we explain the main approaches proposed for topic detection. Some works may exploit more than one category of approaches and so have a hybrid approach. The following study does not, however, focus on these approaches. In general, topic detection can be classified into two different types: New Event Detection (NED) and Retrospective Event Detection (RED). In NED, streaming data is analyzed in order to detect new events (topics), whereas RED aims at identifying previously unknown events from historical collections and is considered as an offline task. Here, examples of both NED and RED analysis in different categories of topic detection approaches are presented. Nevertheless, since the target data in our problem is non-streaming, we mainly focus on RED type of topic detection.

3.2 Topic Detection and Tracking

Topic Detection and Tracking (TDT) (Allan, 2002), pursued since 1997, is an integral part of the DARPA Translingual Information Detection, Extraction and Summarization (TIDES) program. TDT is one of the main projects in topic detection, which analyzes streaming data from news wires. The goal of the project is to help news analysts to detect and follow new and trendy events. This category of approaches, however, does not fit our type of data, which is non-streaming. Consequently, some of the main steps in the TDT approaches, that will be explained in the following, are not required for our problem.

In the procedure of TDT, the text is initially transformed into individual news stories. The stories are then analyzed for events that have not been seen before. Stories which are discussing the same news topic are finally grouped together. According to the definitions in TDT, "story" is defined as a section of transcribed text with substantive information content and a unified topical focus. "Event" and "topic" are then respectively defined as "a specific

thing that happens at a specific time and place along with all necessary preconditions and unavoidable consequences" and "an event or activity, along with all directly related events and activities" (Allan, 2002).

More specifically, TDT consists of two tasks: *topic/event detection*, where new topics/events are detected from the corpus, and *topic tracking*, which tracks evolution of existing topics over time. Depending if streaming or historical data is analyzed, the topic detection step can perform NED or RED for detecting events. In NED, for a newly arrived document, the similarity between the document and the known events is computed. If the maximum similarity is more than a predefined threshold, the document is considered to be related to the corresponding event. Otherwise, it is detected as a new event. Different measures can be used to find the distance between a document and an event. Allan et al. (1998) used a modified version of TF.IDF along with the time distance between the documents, *i.e.* the arriving story and the ones belonging to the known events. Estimating IDF on a streaming data is, however, not a trivial task. Allan et al. (1998) exploited an auxiliary dataset to estimate IDF, while Yang et al. (1998) proposed an incremental IDF factor.

There have been studies in the literature for detecting retrospective events in RED task. Yang et al. (1998) introduced augmented Group Average Clustering (GAC), an agglomerative clustering which is able to detect retrospective events. Li et al. (2005) detect such events by proposing a probabilistic model and using Expectation Maximization (EM) to maximize the likelihood of the distributions. However, their approach requires *a priori* knowledge about the number of events, which is not always an available piece of information.

TDT has become a baseline for evaluating many approaches which aim at analyzing streaming data (Kumaran and Allan, 2004, Lavrenko et al., 2002, Makkonen et al., 2004, Phuvipadawat and Murata, 2010). This project ended in 2005, but the provided data is still available and is widely used as benchmarks in the domain of topic detection.

3.3 Topic Modeling

Natural language text has a rich structure. Words are combined to generate phrases and in a higher level to generate sentences, which eventually make the whole text. The meaning behind any text is inferred by means of its constituent units and their relations within the text. One of the main goals of natural language processing is to infer the semantics of a given text. Some works make use of topic modeling methodologies to detect the underlying topics in a text and to consequently infer its semantics. These methodologies can be considered as the next generation of TDT approaches.

A *topic model* is a type of statistical model, which takes a collection of texts and discovers a set of topics and the degree to which each document covers those topics. The process of discovering the hidden (latent) topics is then referred to as *Topic modeling*. There are two core assumptions behind topic modeling: 1. A document contains a mixture of topics with various proportions, 2. A topic can be approximately described by a set of

words. One of the most significant applications of topic modeling is document classification and retrieval. So far, different kinds of documents have been analyzed using topic models, *e.g.* websites for spam filtering (Bíró et al., 2008), emails for generating summary keywords (Dredze et al., 2008), scientific abstracts (Blei et al., 2003, Griffiths and Steyvers, 2004), and newspaper archives (Wei and Croft, 2006).

3.3.1 Latent semantic analysis

Latent Semantic Analysis (LSA) (Landauer et al., 1998), also named Latent Semantic Indexing (LSI), is an early topic model that was initially patented in 1988 (US Patent 4,839,853). LSA assumes that the semantic similarity between two given words can be inferred from their usage in a text. LSA analyzes the latent (hidden) semantics in a corpora of text with the goal of determining the relationship between terms and concepts in the corpus and comparing texts using a vector-based representation that is learned from the corpus.

Using Singular Value Decomposition (SVD) (Golub and Reinsch, 1970), which is a dimensionality reduction technique, documents and terms are mapped into a lower dimensional space that is generated based on word co-occurrence in the collection of documents. Documents or terms with closer semantic meanings tend to be closer in the semantic space.

To simplify the problem, in LSA, documents are represented as a bag-of-words model, where the order of the words within the document does not matter and only their co-occurrence is taken into consideration.

3.3.2 Probabilistic topic models

Probabilistic topic models are stochastic models for topically annotating a large collection of documents (Steyvers and Griffiths, 2007). They are also generative models for documents. In these models, a document is generated using a probabilistic procedure which chooses a distribution over topics and randomly selects each word of the document from the topics of the distribution. The generative process of probabilistic topic models contains both observed and hidden variables. This process defines a joint probability distribution over both types of variables. Using the joint distribution, conditional distribution of the hidden variables given the observed ones can be calculated. The generative process can be also inverted by obtaining a probability distribution over topics using statistical inference when a document is given. Figure 3.1 illustrates both the generative process and the statistical inference in probabilistic topic modeling.

Hofmann (1999) introduced probabilistic latent semantic indexing (PLSI), also known as probabilistic latent semantic analysis (PLSA) and the aspect model. PLSI is a probabilistic topic model released after LSI method to fix some of its limitations. In PLSI, each word of a document is modeled as "a sample from a mixture model, where the mixture components are multinomial random variables that can be viewed as representations of topics" (Blei et al., 2003). In other words, each word is generated from one topic and

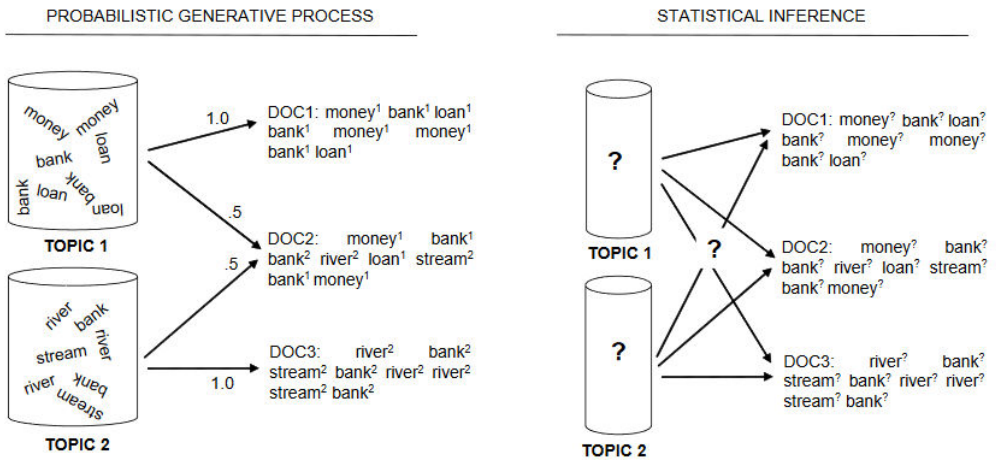


Figure 3.1: Illustration of the generative and the statistical inference problems (Steyvers and Griffiths, 2007)

hence each document in the corpus can be seen as a probability distribution on a fixed set of topics, which provides a more abstract view over its content.

Unlike LSI, PLSI distinguishes different meanings of polysemous words without making use of any dictionary or thesaurus. In addition, PLSI's results are easier to interpret due to the well-defined probabilities.

According to Blei et al. (2003), PLSI is not a well-defined generative model as it cannot assign probability to an unseen document out of the training set. Furthermore, the number of parameters which must be estimated in PLSI grows linearly with the number of the training documents and this leads to overfitting problems. Latent Dirichlet Allocation (LDA) (Blei et al., 2003) overcomes both of these problems by treating the topic mixture weights as a k-parameter hidden random variable rather than a large set of individual parameters which are explicitly linked to the training set. LDA is the most widely used topic model nowadays. It is based on the assumption that documents are represented as random mixtures over latent topics, where each topic is a distribution over words. Unlike PLSI, LDA can describe the probabilistic procedure for both observed and unseen documents.

LDA is used in some applications of natural language processing and information retrieval, such as recommendation systems, document clustering, role discovery in social networks, etc. However, it suffers from some limitations. In this model, the number of topics is fixed and must be known *a priori*. Obviously, for many datasets, estimating the number of topics without any knowledge about the dataset is not a trivial task. Some heuristics attempt to find the optimal number of topics. As an example, Teh et al. (2004) determine the number of topics by the collection during posterior inference. However, there is no silver bullet for this problem and estimating the number of topics brings additional complexity to topic detection approaches. Moreover, in LDA, topics are not correlated, since the Dirichlet topic distribution cannot capture correlations. Topics detected using LDA are also static and cannot evolve over time. This, however, is not an issue for non-

streaming corpora. The next limitation of LDA is that it requires many documents for the learning phase. More specifically, it learns topics by sampling documents of the training set over and over again. Hence, the more documents are available, the more accurate the result will be. This property imposes a limitation for applying LDA on applications with only few available documents.

As LSI, both PLSI and LDA use the bag-of-words model. Unlike these approaches that perform word-level analysis, we aim at targeting both single and multi-token keywords. Hence, in our analysis, the order of words in documents is important as the multi-token keywords are generated based on it. We, however, can name our model a "bag-of-keywords" model, since after generating the keywords, their order does not matter in the topic detection phase.

Since 2003, there has been extensions to LDA in order to overcome some of its limitations. Pachinko allocation model (PAM) (Li and McCallum, 2006) and the correlated topic model (CTM) (Blei and Lafferty, 2007) are able to discover the connections between topics, which have been missed in the original LDA model. Identifying two highly correlated topics, one can relate both of them to a given document even if only one of the topics has been explicitly discussed in the document. However, although CTM generates more interesting results than LDA, it is a more computationally expensive model. Another important extension of LDA is the dynamic topic model (DTM) (Blei and Lafferty, 2006), which aims at capturing evolutionary topics by dividing the data into different time slices and modeling documents of each slice separately. The topics in time slice t are the evolved topics from time slice *t-1*. Blei et al. (2004) also proposed an extension of LDA for modeling a treelike hierarchy of topics, where the lower levels of the tree represent more generic topics, while the higher levels show more specific and more fine-grained topics. Unlike this work in which each document must select topics from a single path in the tree, nested hierarchical Dirichlet process (nHDP) (Paisley et al., 2015) allows the documents to access the entire tree by defining priors over a base tree.

As other extensions of LDA, Reisinger et al. (2010) proposed spherical topic model, which assigns positive and negative weights to a topic terms. Unlike the positively weighted terms, the negatively weighted ones are not related to the topic and are very unlikely to appear in the documents which discuss the topic. Furthermore, Doyle and Elkan (2009) proposed a topic model, which uses Dirichlet compound multinomial (DCM) distributions in order to model the burstiness phenomenon in the word counting phase to make it more realistic. Based on the burstiness phenomenon, if a word appears once, it is more likely to appear again. A fine-grained list of topics has been also identified by Xie and Xing (2013). In their work, the document clustering is used to identify local topics, specific to each group of documents, and global topics, shared by all groups.

The original LDA is an "unsupervised" model in which latent topic variables have "directed" connections to observed ones, which represent words in a document. By contrast, some topic models make use of the metadata in the collection documents and so are "supervised" (Lacoste-Julien et al., 2009, Mcauliffe and Blei, 2008, Ramage et al., 2009, Rosen-Zvi

et al., 2004, Zhu et al., 2010). Moreover, some approaches aim at developing topic models using "undirected" graphical models (Gehler et al., 2006, Hinton and Salakhutdinov, 2009, Xing et al., 2005).

Recently, there are approaches which make use of word embeddings in topic modeling to get more coherent topics. In these approaches, a document is treated as a collection of word embeddings. Topics are then regarded as a distribution in the embedding space. Das et al. (2015) use multivariate Gaussian distribution for this purpose, while Batmanghelich et al. (2016) use Mises-Fisher (vMF) distributions, which rely on the cosine similarity instead of euclidean distance between the word vectors.

Since single words in topics could be difficult to understand and in general compound words contain more information, there are approaches which extend the single words of LDA to compound words. Wallach (2006) developed a bi-gram topic model to consider the dependencies between consecutive words by exploiting bi-gram statistics in the latent topic variables. Wang et al. (2007) later proposed the topical n-gram model, based on the Wallach's bigram model, to target n-grams. Some other works also go beyond n-grams by extracting phrases from the given collection and exploiting them in the topic model in order to generate richer and more interpretable topics (He, 2016, Lindsey et al., 2012, Yu et al., 2013).

Although topic modeling approaches have been widely used in different applications, we believe that words in the detected topics are not always semantically consistent. Some words may share only the notion of co-occurrence without sharing semantics. As an example, having the output of LDA in Figure 3.2, in the "Arts" topic, there is no semantic relation between "NEW", "MUSICAL", and "LOVE". Instead, these words are only expected to appear within the same context. Since in our enrichment problem, semantically consistent topics are required, we do not exploit topic modeling approaches in our problem.

3.4 Graph-based topic detection

Another category of topic detection approaches makes use of the graph structure to show the relation between documents or their (key)words and to further cluster them into different groups. Generally, the graph structure is a way to represent complex information about entities and their interactions. According to Bekoulis and Rousseau (2016), comparing to the standard vector of frequencies used in topic modeling approaches, such as LSA and LDA, the graph structure and effective graph analysis methods can reveal more information about entities' relations. Sayyadi and Raschid (2013) also point out that since topic modeling methods are mainly based on words distributions, they do not explicitly consider word co-occurrences. Alternatively, graph-based models with more explicit relations can be used and consequently more information about entities' relations can be retrieved.

In the graph-based model, topic detection is regarded as a *graph clustering* task, where each detected cluster is considered as a "topic". In each topic, nodes tend to be highly connected and they share significant semantic similarity. Relations among the topics and their

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Figure 3.2: Example of the LDA result (Blei et al., 2003)

overall structure can be obtained from the connectivity between the clusters. Depending on the type of the target data, a graph-based topic detection approach can perform NED or RED.

Since the graph-based model can better disclose latent relations than the topic models, in this thesis, we exploit it for detecting topics with different levels of granularity. The graph-based model can be used along with other approaches of topic detection, such as topic modeling, to increase their performance (Bekoulis and Rousseau, 2016, Zhang et al., 2016). In the following, however, we present the approaches which merely use graph-based models for topic detection.

Graph-based approaches consist of two main steps: 1) graph generation, including node and edge generation phases, and 2) graph analysis. In the following, we explain each step in more details.

3.4.1 Graph generation

Graph generation consists of two main steps: node generation and edge generation. Depending on the target application, one might be interested to have a directed graph in order to model the direction of the relations between the nodes. In some other approaches, however, the relationship is symmetric and an undirected graph can be used to model

the connectivity. The relations between nodes can have different levels of strength or can be treated all the same. These properties are modeled, respectively, using a weighted and unweighted graph. In more complex graphs, nodes can be also weighted to show their different levels of importance. Sayyadi and Raschid (2013) generate an undirected and weighted graph for topic detection. We, however, use an unweighted graph for this purpose.

Node generation. To generate a graph, initially, types of nodes need to be determined. Although nodes can be of various types, for the topic detection problem, they are mostly single words or phrases extracted by one of the methods presented in Chapter 2. As an example, Sayyadi et al. (2009) proposed a graph-based approach to detect new events (topics) and to track stories in social networks. For this purpose, they build a co-occurrence graph of keywords. More specifically, to generate nodes of the graph, a set of keywords, containing both single and compound words, is extracted from the collection of documents and a filtering phase is performed to filter out the keywords with low document frequency. Remaining keywords are then considered as nodes of the graph.

In spite of the wide use of words and phrases as graph nodes, some approaches, such as that of (Garza Villarreal and Brena, 2011), build a graph of documents, where edges show their connection, *e.g.* hyperlinks. By clustering this graph, topically related documents are grouped into different clusters. A summary over the detected topics can be also provided by extracting keywords of the documents.

In this thesis, the ultimate goal is to enrich the content of an unstructured document using a set of keywords. As Sayyadi et al. (2009), we generate a graph of keywords for achieving this goal. The graph model captures the semantic relation among keywords, which can be both single and multi-token.

Edge generation. Generating edges of a graph is basically the most challenging step of the graph generation. Edges determine the type of information that the graph represents. In general, edges of a graph can represent different types of relations between nodes. They can show the links between the graph nodes based on their connectivity in the corpus. An example is hyperlinks between web pages in a graph of web pages. It should be however noted that in some collections, such as a website, some of the hyperlinks between documents are not qualified. In other words, documents might be incorrectly linked together due to human mistakes or spamming techniques. Using the information of these unqualified links, noisy edges are added to the graph. Hence, it will be more accurate to first detect and remove the unqualified links from the collection (Qi et al., 2007).

In case of having words or phrases as nodes of a graph, edges can be generated based on different similarity types, which will be explained in more details in Section 3.7.

3.4.2 Graph analysis

After generating the graph, some analysis should be performed on it in order to obtain the required information. In the topic detection domain, a typical analysis is graph clustering, which clusters the graph into different components. According to Schaeffer (2007), *graph clustering* (also called *community detection problem*) is defined as "the task of grouping the vertices of the graph into clusters taking into consideration the edge structure of the graph in such a way that there should be many edges within each cluster and relatively few between the clusters". Each group of vertices is called a *cluster* or a *community* that in the domain of topic detection is referred to as a *topic*. Graph clustering should not be however confused with *graph partitioning* which divides vertices of the graph into a pre-defined number of groups "such that the number of edges lying between the groups is minimal". Due to the necessity of providing the number of groups as an input parameter, graph partitioning algorithms are not good for community detection, where no *a priori* knowledge is available about the number of communities.

Some algorithms of community detection support overlapping nodes between communities, whereas some others generate a set of disjoint communities. A comprehensive review over community detection algorithms has been done by Fortunato (2010). Depending on the target application, one might be interested to have disjoint topics, where each term belongs to only one topic. Disjoint topics, however, do not support polysemous terms. Since most of the real-world graphs contain overlapping communities, we focus on overlapping community detection algorithms in this thesis. The main algorithms for detecting overlapping communities have been reviewed by Xie et al. (2013).

Clique Percolation Method (CPM) (Palla et al., 2005) is the most acknowledged algorithm for detecting overlapping communities through node partitioning. The main assumption behind this algorithm is that cliques are more likely to be formed by the internal edges of a community rather than the inter-community ones. One limitation of CPM is that vertices with degree of one, *i.e.* leaves of the graph, will not belong to any community and are missed in the process of community detection. This algorithm returns communities for various clique sizes. However, automatically detecting the best size of the cliques, which returns the most meaningful structure, is not a trivial task and this can be considered as another limitation of CPM. The original algorithm of CPM is applicable on an undirected and unweighted graph but it has been extended to directed and weighted graphs (Farkas et al., 2007, Palla et al., 2007). A fast implementation of CPM has been proposed by Kumpula et al. (2008).

A group of overlapping community detection algorithms partitions links rather than nodes to obtain communities (Ahn et al., 2010). In these algorithms, a node is an overlapping node if its connected links belong to more than one cluster. Some approaches perform link partitioning by transforming the original graph into a line graph, where nodes are the links of the original graph (Evans and Lambiotte, 2009, 2010). However, Fortunato (2010) states that there is no guarantee that the link partitioning algorithms outperform node partitioning ones. According to Ding et al. (2016), node clustering algorithms need prior

information for performing community detection. Examples of this information is the number of the communities, the size of the cliques, etc. Due to the high complexity of link clustering approaches, authors propose an approach which detects overlapping communities based on network decomposition.

In spite of the disadvantages of node clustering algorithms, due to their lower complexity, they are widely used for community detection task. As an example, Sayyadi and Raschid (2013) make use of the betweenness centrality measure to detect communities in their generated graph. Authors point out that their proposed graph-based approach, named KeyGraph[1], is the first attempt to consider explicit co-occurrence between terms. In their approach, edges of the graph are generated according to the co-occurrence of the nodes within given documents. To avoid noisy data, authors remove nodes with low document frequencies. Low-weighted edges, which correspond to nodes with low co-occurrences, are also removed. Two conditional probabilities are computed for each edge. If the values of both probabilities are lower than a pre-defined threshold, the corresponding edge is also filtered out. Authors then exploit the betweenness centrality measure for the graph analysis step, assuming that inter-community edges have higher betweenness scores, since the shortest paths between nodes from different communities pass through them. Communities of the graph are detected by removing the highly scored edges. The performance of KeyGraph was compared with k-Nearest Neighbor (kNN), Augmented Group Average Clustering (GAC) and Latent Dirichlet Allocation (LDA) on TDT4 benchmark. Results showed that KeyGraph's performance is comparable to that of GAC and LDA with Gibbs sampling. Time complexity of KeyGraph is, however, considerably lower than LDA. More specifically, their results show that KeyGraph outperforms GAC and LDA for precision value. However, LDA performs slightly better than KeyGraph and GAC in terms of recall and F-measure. Table 3.1 shows the values obtained by each of these three methods on TDT4 benchmark.

Table 3.1: KeyGraph performance with respect to the state-of-the-art methods on TDT4 benchmark

Method	*Precision*	*Recall*	*F-measure*
KeyGraph	**0.82**	0.59	0.69
GAC	0.79	0.59	0.68
LDA	0.8	**0.61**	**0.7**

Similarly, we make use of a node partitioning community detection algorithm to detect overlapping communities. More specifically, we exploit CPM for graph clustering. The main motivation behind using this algorithm is to detect overlapping communities (topics) and so to let the keywords belong to more than one topic. The limitation of missing the graph leaves in CPM is not an issue for our application, as these nodes are mostly not semantically related to any detected topic and removing them from the result is an

[1]This should not be confused with the "KeyGraph" approach proposed by Ohsawa et al. (1998b). The similarity between these two approaches is that they both generate a graph of extracted terms based on their co-occurrences in a given text. The way to extract terms is, however, different in the two works.

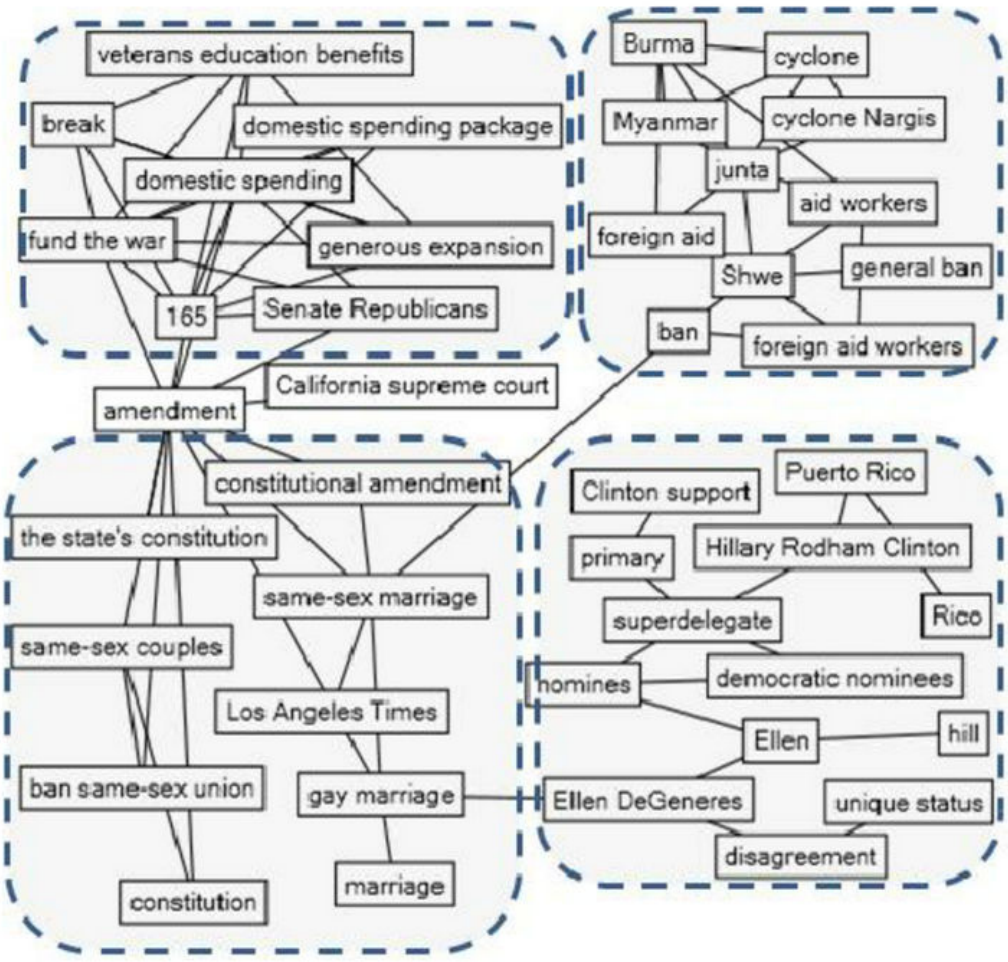


Figure 3.3: Example of the KeyGraph result (Sayyadi et al., 2009)

advantage for our approach. To overcome the difficulty of finding the best clique size, we propose an algorithm that goes through different values of clique size and returns the best communities (see Subsection 7.1.2). In other words, we will show that there is not a single value of clique size that returns the best communities. Hence, in our approach, the final communities can be a mixture over the communities returned for different sizes of cliques.

We note that comparing to Sayyadi and Raschid (2013), we aim at detecting more semantically consistent topics. Figure 3.3 shows an example of their result. For our enrichment problem, we did not find their detected topics to contain only semantically related keywords and this is due to the notion of co-occurrence that they exploit in their approach. As an example, in the top left community, "165" does not share any semantics with the other keywords of the community. Hence, we make use of a graph-based model but we model the connectivity between keywords based on the semantic relatedness and not the explicit co-occurrence in studied documents.

3.5 Evaluation of topic detection

Topic detection approaches are evaluated in different ways. Evaluation of topic detection is mainly characterized by the exploited benchmarking data and the evaluation measures. Here, we briefly explain the main benchmarks used in this domain along with the most widely-used evaluation measures.

3.5.1 Benchmarks

The general procedure for evaluating topic detection approaches is to compare their detected topics with the gold standard topics associated to a benchmarking data. Depending on the type of input data and the target application, one can use an existing benchmark or generate an application-specific one. One advantage of using an existing benchmark is that it provides a way to compare the proposed approach with different approaches in the literature.

TDT benchmarks are one of the most widely-used benchmarks in topic detection domain. Linguistic Data Consortium (LDC) released five TDT benchmarks in different years, which contain a great number of news stories. Each story has been labeled manually for the relevancy to a set of pre-determined topics. In addition to TDT, more specific benchmarks can be used for evaluation. As an example, Wartena and Brussee (2008) performed their proposed model on Wikipedia articles and considered 8 categories of Wikipedia as the benchmark for evaluation.

In spite of the wide use of the TDT benchmarks, we do not use them to evaluate our topic detection approach. Using TDT benchmarks requires an approach for extracting keywords from news stories. Our keyword extraction approach, however, has been proposed for web pages and it cannot effectively extract keywords from news stories. Hence, to perform the proposed topic detection approach on news stories, their corresponding keywords must be given as an input and if this data is unavailable, our proposed topic detection approach cannot be evaluated on this benchmark. As will be explained in Chapter 8, we create our own experimental data, which consists of sets of keywords extracted from web pages using our proposed keyword extraction approach.

3.5.2 Evaluation measures

Effectiveness of an approach of topic detection is determined through the match between its detected topics and the ones in a gold standard set or in a benchmark. This effectiveness is represented by means of evaluation measures. The traditional measures for this purpose are *precision*, *recall*, and *F-measure*. Basically, precision and recall values are computed individually for each detected topic. To have the overall evaluation, per-topic precision and recall values are combined in different ways. The straightforward approach is to take an average over them and to calculate F-measure on the averaged precision and recall.

In graph-based topic detection approaches, the problem of evaluating the detected topics is regarded as the problem of evaluating the communities detected using a community detection approach. There are different measures for evaluating disjoint communities. However, as previously mentioned, most of the real world communities overlap. According to Xie et al. (2013), only few measures can be used for evaluating overlapping communities and among them Normalized Mutual Information (NMI) and Omega Index are the most widely-used ones. NMI has been extended by Lancichinetti et al. (2009) in order to be used for evaluating overlapping communities and Omega Index (Collins and Dent, 1988) is the overlapping version of the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985). As will be seen in Chapter 8, we do not exploit these measures due to the difficulties of generating a gold standard set of topics. Instead, we perform a user-based evaluation and evaluate the effectiveness of our approach using the traditional measures, *i.e.* precision, recall, and F-measure.

3.6 Conclusion on topic detection approaches

So far, many topic detection approaches have been proposed and only the main categories of these approaches are presented in this thesis. Depending on the type of the input data and the target application, the choice of the topic detection approach can be different. Topic Detection and Tracking (TDT) approaches and their extensions are applied on streaming data. If a corpus of documents is available, topic modeling approaches can be used to learn term relations from the corpus. This relation can reveal both the co-occurrence and the semantic relatedness between the terms. In case of having phrases as terms or having no prior knowledge about the number of the topics in the corpus, more advanced topic models must be employed. To model more latent relations and to target mostly the semantic relation between terms, graph-based approaches with semantic edges are a better choice. However, in spite of the advantages of the graph-based models, there is a lack of research in this category of approaches. To the best of our knowledge, there is no domain-independent graph-based approach of topic detection, which would be easily tunable to different languages and merely return semantically consistent topics including both single and multi-token keywords.

3.7 Terms similarity

In this work, we aim at enriching a document using keywords, which are semantically related to its content. Therefore, in our graph-based approach of topic detection, measures for capturing semantic relatedness need be exploited in order to connect the graph nodes accordingly.

The semantic notion between two terms can be referred to as *semantic similarity* (*semantic distance*) or *semantic relatedness*. It should be noted that semantic relatedness is a broader notion than semantic similarity and it covers more types of relations. While rela-

tions such as synonymy can be captured using semantic similarity, other types of relations, such as meronymy, hyponymy, and antonymy, are captured using semantic relatedness. In our content enrichment problem, we are interested in capturing semantic relatedness and not merely semantic similarity. Types of relations in semantic relatedness, however, can vary from one application to the other. In our thesis, we aim at capturing all the synonymy, meronymy and hyponymy relations between keywords. It should be noted that antonyms are not relevant here, as we are not interested to enrich a content using them.

In the following, we present the most-widely used categories of similarity: *morphological* and *semantic*. These categories can be also combined together as a *hybrid* similarity. Although capturing semantic similarity is the main challenge in our work, some steps of the topic detection approach exploit a hybrid similarity to better model the connectivity between keywords (Section 7.2). To effectively capture the semantic between any two keywords, new semantic similarities are proposed (Section 7.1).

3.7.1 Morphological similarity

Morphological similarity aims at finding the string-based similarity between terms. One category of this similarity depends on the number of common tokens between any two terms. Cosine similarity, Jaccard index, Dice's coefficient, Longest Common SubString (LCS), N-Gram similarity, etc., are all examples of this category. There is also another category of morphological similarity, which is based on transformations between strings. Levenshtein distance (Levenshtein, 1966) is a typical example of this category. In practice, morphological similarities fail to capture the relation between terms or in general between strings with no/not enough common tokens or characters. Due to this limitation, they cannot effectively represent the "semantic" similarity between terms.

3.7.2 Semantic similarity/relatedness

According to Gomaa and Fahmy (2013), measures which capture semantic similarity and semantic relatedness are grouped into two categories: *knowledge-based similarity* and *corpus-based similarity*. In the following, we explain each of these categories.

Knowledge-based similarity

Using the information derived from semantic networks, knowledge-based similarity is capable of revealing both the semantic relatedness and the semantic similarity between terms. WordNet (Miller et al., 1990) is an example of a semantic network that has been widely used for calculating semantic similarity between pairs of words (Corley and Mihalcea, 2005, Kamps et al., 2004, Patwardhan, 2006, Richardson et al., 1994, Wan and Angryk, 2007).

The semantic similarity measures used in this category have been proposed by Jiang and Conrath (1997), Leacock and Chodorow (1998), Lin (1998), Resnik (1995), Wu and

Palmer (1994). The semantic relatedness between terms is also computed using measures proposed by Hirst and St-Onge (1998), Lesk (1986), Patwardhan (2003).

Generating and maintaining a knowledge base is not a trivial task. The existing knowledge bases may also contain outdated information. New words added to a language and their relations with other words may not be covered by the existing knowledge bases. Due to these limitations, we do not exploit any knowledge base in our approach. In addition, we aim at applying our topic detection approach on domain specific keywords and hence we require a method that can be effectively applied on any domain. Such a method needs to be knowledge poor as we cannot have a relevant knowledge base for each new domain of interest.

Corpus-based similarity

Corpus-based similarity aims at capturing the semantic similarity and the semantic relatedness between terms using the models of information theory learned from large text collections. Comparing to knowledge bases, the content of text corpora is updated more regularly with much less effort. Hence, it better covers new words and relationships. Approaches proposed by Islam and Inkpen (2006), Kolb (2009), Lund and Burgess (1996), Turney (2001) and topic modeling approaches, such as LSA and LDA, make use of this category of similarity.

Some approaches use web data as corpus. The motivation behind using the web is that it is a huge and multilingual resource, which is written and updated regularly by different people. In addition, it covers any domain. New words are also added to the web frequently. Hence, it is a good resource for mining the semantic relationship between unseen words. The web contains common words, found in news articles, forums and blogs, and also specific terms, found in scientific documents. Nevertheless, compared to some other corpora, such as a collection of scientific papers, the web content is written by different types of people, including experts, non-experts, volunteers, etc. Hence, it is more likely to encounter noise in this corpus. In addition, not all the content provided on the web is informative. Some web pages contain spam content or wrong information. To mitigate these drawbacks, a web-based approach must be robust to the noise in the web content. In addition, the most relevant and the most important web pages within the domain of study must be considered as the corpus.

Due to its advantages, web content has been used as source of information for many NLP applications. Lin and Zhao (2003) use web content in order to identify synonyms among distributionally similar words. Zhu and Rosenfeld (2001) improve trigram language modeling using the n-gram counts returned by search engines. The tremendous data on the web is used by Dumais et al. (2002) to generate a big enough training dataset in order to increase the accuracy in question answering task. More specifically, web-based semantic similarity has been used in different applications, including automatic annotation of web pages (Cimiano et al., 2004), extraction of the underlying relations between entities in

social networks (Mori et al., 2006), predicting the genre of a given artist (Schedl et al., 2006), etc.

Although some works exploit search engine-independent web documents as source of information, such as that of (Sheetal A Takale, 2010) which uses Wikipedia, some approaches make use of search engines results and/or their functionalities, such as page counts. Due to the efficiency of search engines in returning highly relevant documents as a response to a query, we believe that their information can be exploited in order to overcome the unreliability problem of web pages. In other words, since search engines give higher ranks to web pages with richer and more relevant content, we assume that they can generate a more reliable corpus. Hence, in order to benefit from the advantages of web data and to overcome its unreliability drawback, we make use of the web content provided by search engines as the corpus for computing similarity in our topic detection approach. A search engine-based approach needs to be optimized in terms of the number of requests sent to the search engine. Since even the highly ranked pages may contain noise, our topic detection approach must nevertheless be robust to the noise in the content of web data.

Normalized Google Distance (NGD) (Cilibrasi and Vitanyi, 2007) is one of the most acknowledged measures which computes the semantic similarity using Google search engine. In this measure, hits values, returned by Google, are used as a way to model the co-occurrence between two given terms that are sent as queries to the search engine. *Hit* or *page count* is the number of pages that contain the query words. Equation 3.1 shows the formula for calculating this measure, where N is the total number of the indexed pages by Google and a and b are the terms (queries) for which the NGD measure is computed.

$$NGD(a,b) = \frac{max(log(hit(a)), log(hit(b))) - log(hit(a,b))}{log(N) - min(log(hit(a)), log(hit(b)))} \tag{3.1}$$

In another work, Bollegala et al. (2007) exploit both page counts and text snippets returned by search engines. The former information is used for calculating four different similarity scores, which are web-based versions of Overlap coefficient, Jaccard, Dice and Pointwise Mutual Information (PMI) measures (Equations 3.2 to 3.5). Authors, however, did not find the page counts to be enough for calculating the semantic similarity and also exploited the content of snippets. They automatically extract lexico-syntactic patterns from these contents to show the relation between queries. Examples of these patterns are *also known as, is a, part of*, etc. Two-class support vector machines (SVMs) are then used to integrate the page counts-based and the pattern-based measures and to get the final measure of similarity.

$$WebJaccard(a,b) = \begin{cases} 0 & \text{if } hit(a \cap b) \leq c \\ \frac{hit(a \cap b)}{hit(a)+hit(b)-hit(a \cap b)} & \text{otherwise.} \end{cases} \tag{3.2}$$

$$WebOverlap(a,b) = \begin{cases} 0 & \text{if } hit(a \cap b) \leq c \\ min(hit(a), hit(b)) & \text{otherwise.} \end{cases} \tag{3.3}$$

$$WebDice(a,b) = \begin{cases} 0 & \text{if } hit(a \cap b) \leq c \\ \frac{2hit(a \cap b)}{hit(a)+h(b)} & \text{otherwise.} \end{cases} \tag{3.4}$$

$$WebPMI(a,b) = \begin{cases} 0 & \text{if } hit(a \cap b) \leq c \\ \log(\frac{\frac{hit(a \cap b)}{N}}{\frac{hit(a)}{N}\frac{hit(b)}{N}}) & \text{otherwise.} \end{cases} \tag{3.5}$$

Text snippets have been also used in other works in order to calculate the similarity between two given terms/queries. Sahami and Heilman (2006) expand the vocabulary of two given queries using the snippets returned for those queries by a search engine. Each snippet is presented as a TF.IDF vector and each vector is L2 normalized in order to calculate the centroid of the set of vectors. Semantic similarity between two queries is then defined as the inner product between the corresponding centroid vectors. In the same way, Chen et al. (2006) collect the snippets related to two queries. They then find the similarity of the queries using a double-checking model, where occurrences of the first query in the snippets returned for the second query (forward process) and the reverse procedure (backward process) are counted. Considering the two values, authors tried different formulae for calculating the association scores between the two queries. They also showed that the best performance of the measure is obtained by exploiting 600 snippets.

Iosif and Potamianos (2010) proposed a context-based similarity measure and showed that it outperforms co-occurrence-based similarity. They rely on contextual information in the documents returned by search engines. For a target word, a vector is generated, which represents the frequency of each word of the vocabulary within its left or right context (with a pre-defined window size). The similarity between two words is then measured by computing the cosine similarity between their corresponding vectors. Similarity within word-groups is calculated using search counts in an approach proposed by Gledson and Keane (2008).

As mentioned before, in our topic detection approach, we propose two similarities in order to capture the semantic relatedness between graph nodes. These measures are close to the ones proposed by Sahami and Heilman (2006) and Chen et al. (2006) but the details of our measures and also the exploited corpus are different in our work. In Chapter 7, we will discuss that the page count-based measures do not work anymore due to the changes in search engines policies and functionalities. In general, our proposed similarities use the context of SERP but do not rely on their functionalities, such as page counts. Similarities which exploit these functionalities are likely to perform effectively only for a specific period of time.

3.7.3 Hybrid similarity

Hybrid similarities take advantage of different types of similarities for capturing the relation between terms. A hybrid similarity measure has been proposed by Mihalcea et al. (2006), where both corpus-based and knowledge-based measures are used for calculating similarity.

Authors show that the similarity is better detected when both types of similarities are exploited. A combination of corpus-based and knowledge-based measures has been also used by Aggarwal et al. (2012) in order to find the semantic similarity between sentences. Islam and Inkpen (2008) proposed another hybrid similarity, which combines semantic and syntactic information to measure the similarity between two texts. Authors exploited both a corpus-based measure and a normalized and modified version of the Longest Common Sub-sequence (LCS) string matching algorithm for this purpose. Buscaldi et al. (2012) also combined structural and conceptual similarity measures to calculate the similarity between concepts. The former measure exploits an n-gram based similarity between sentences and the latter makes use of WordNet.

In our work, to detect the underlying topics within a collection of keywords, we only make use of corpus-based semantic similarity. However, to further divide the topics and to generate a set of fine-grained topics, we exploit a hybrid similarity, consisting of a morphological and a corpus-based semantic similarities (see Section 7.2).

Part III

Methodology

Chapter 4

The overall methodology

Contents

4.1 Refined problem statement

Writing the content of a document with respect to a domain is a challenging task as one needs to make sure that the generated document contains the vocabulary and the information that is commonly used and highly discussed in the domain. The task is becoming more challenging in competitive environments, where a document must not miss such a vocabulary or information when compared to other thematically relevant documents. Considering an input document and a domain of interest, quite often, there is a gap between them. The gap, called *semantic gap* in this thesis, can be vocabulary-based and/or informational. In the former case, the input document and the target domain deal with the same topic but do not use the same words, while in the latter case, they deal with connected or overlapping topics but the document is missing part of the information of the domain. To minimize the semantic gap between an input document and a target domain, we need an approach to automatically analyze the documents in the domain and to return their vocabulary or information, which can be used for enriching the content of the input document.

In this thesis, we aim at addressing this problem: enriching an unstructured document with respect to a collection of documents, which is representative of a domain of interest. The goal is to minimize the semantic gap between the input document and the target domain. At the end of the enrichment procedure, a list of keywords is recommended to a user, who aims at enriching a document. These recommended keywords reflect the main information of the collection documents. The collection may, however, deals with various topics. Hence, we need to detect their underlying topics and to recommend only keywords which are related to the target domain.

Although existing approaches of topic detection have been proven to perform well in some cases, they do not meet the requirements of our particular application. Unlike the TDT approaches, we target non-streaming data. Hence, there is no need to deal with the complexities related to the streaming nature of the input data, *e.g.* first story detection, story segmentation, etc.

Topic modeling approaches can be performed on non-streaming data. However, these approaches make use of implicit co-occurrences between the studied keywords and so cannot model all their relations. Studying the topics detected by LDA approach shows that the topic keywords are not necessarily semantically consistent. Hence, they do not meet the requirements of our application, where topics must contain semantically related keywords. In addition, in these models, a topic is represented as a set of words and no label is assigned to it to show its main subject. The words associated to a topic have the same level of importance and representativeness. However, in reality, some words contribute more in the representation of a topic. The major limitation of the widely used topic model, *i.e.* LDA, is that the number of topics must be approximated *a priori* and this is a challenging task when no information about the collection is available. In addition, this approach, originally, does not support multi-token words and employs the bag-of-words model without taking into consideration the order of words.

Even though the graph-based model is able to effectively represent the relations between terms, *i.e.* words or phrases, existing graph-based approaches mainly focus on co-occurrence connectivity rather than semantic relationship between terms. There is also a lack of research in studying the granularity of the topics detected by these approaches. In spite of its advantages, the graph-based model has not been studied enough for the topic detection task and it deserves more attention.

Considering all the above mentioned points, to the best of our knowledge, there is no effective, robust, and domain-independent approach for enriching the content of an unstructured document using keywords that are semantically related to a target domain. This thesis proposes such an approach and specifically focuses on enriching the content of web pages. The goal is to facilitate the analysis of a collection of web pages, to generate more informative content for visitors, and to return web pages more effectively in indexing process so that they can be easily retrieved.

In the following section, we explain our methodology in more details.

4.2 Methodology

In this section, we explain our proposed approach for recommending semantic-oriented information. The ultimate goal is to enrich the content of an unstructured document with respect to a domain of study. The domain is specified by an input keyword and the enrichment information is provided by side documents. In this work, we specifically focus on enriching the content of non-streaming web pages.

The proposed approach requires interactions with a user who performs document enrichment. As input data, the user specifies a document, called *input document*, and a specific keyword, which represents the domain of interest and determines the enrichment's point of view. Specifying the point of view avoids a general enrichment and so gives an advantage to our enrichment approach. In other words, it helps users to focus the enrichment on a specific domain.

The enrichment is done by recommending a list of keywords to the user. Eventually, the user makes the final decision on the recommended keywords and adds the desired ones to the input document. Hence, the final keywords are not added directly to the document and are firstly verified by the user. This manual verification is required in order to avoid adding pieces of information which are related to the target domain but are not interesting to be added to the input document. As an example, considering the domain of "ustensile de cuisine", both "poêle" and "casserole" are detected as relevant keywords. However, an input web page that sells "poêle" may not intend to sell "casserole". In this case, the latter word must be filtered out by the user to not be added to the content of the page.

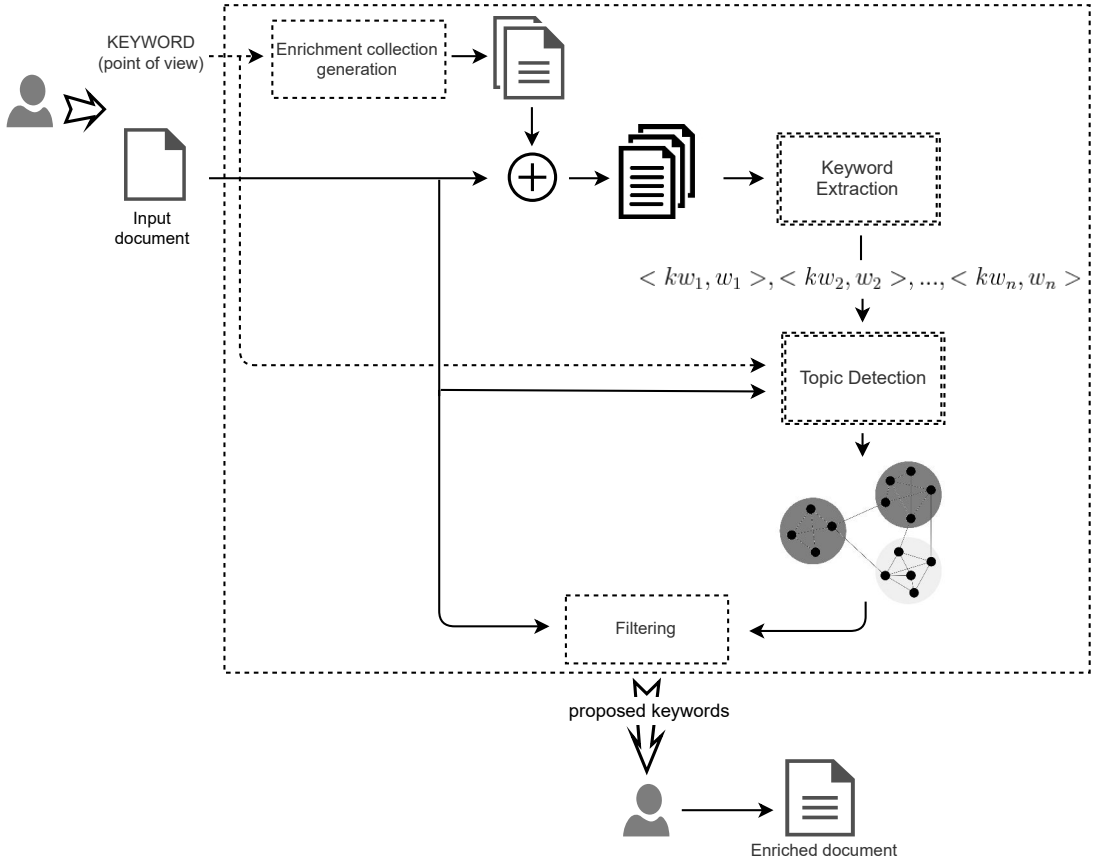


Figure 4.1: Overall framework of the proposed approach. The enrichment point of view is determined by the user as an input keyword.

We consider a collection of documents as representative of the domain of study. To find the enrichment information, we decompose the problem into four steps: *enrichment*

collection generation, *keyword extraction*, *topic detection* and *filtering*. The main steps of the approach are keyword extraction and topic detection. In keyword extraction step, the main information of documents within the enrichment collection is extracted and returned as a ranked list of keywords. Topic detection step then detects the underlying topics in the extracted list. Additional analysis is performed on each detected topic in order to recommend sets of semantically similar keywords. Here, we present the functionality of the proposed approach along with the explanations on collection generation and filtering steps. Keyword extraction and topic detection steps are also briefly explained in this section. More details on these two steps are respectively presented in Chapters 5 and 7.

Figure 4.1 shows the overall framework of our proposed approach. The approach starts by taking a document and an input keyword from a user who performs the enrichment. The input keyword is determined according to the target domain, which specifies the enrichment's point of view. A collection of relevant documents is then generated with respect to this point of view. This collection is considered as a source of information for our enrichment approach and can be generated in different ways depending on the type of documents in the target application. In this thesis, we specifically focus on enriching web pages. Hence, our proposed enrichment collection generation step aims at retrieving web pages related to a given point of view. The input document and the generated collection are both passed to the keyword extraction step, where the main information of the documents is retrieved and represented as a ranked list of keywords. The keywords can consist of one or more tokens. The motivation behind extracting multi-token keywords is to capture the association between words and consequently to extract more informative keywords. Topic detection is then performed on the extracted keywords in order to detect the underlying topics within the collection of keywords. Topics related to the target domain are then returned as "relevant" topics for further analysis.

The first set of detected topics are coarse-grained. More specifically, we detect fewer topics but more generic ones, which are large in terms of the number of constituent keywords. This step is essential for distinguishing different topics in the collection of keywords and also for disambiguating polysemous keywords in the collection. However, due to the generality of the topics, the result of this step is not well-organized and informative enough for recommendation. A more structured representation of topics is required to get focused enrichment information. Hence, additional analysis is performed in order to divide each coarse-grained topic into a set of fine-grained topics. This set is then passed to the filtering step, where keywords which already exist in the content of the input document are filtered out. The remaining keywords are eventually proposed to the user in order of relevancy and importance. Recommending an ordered set of keywords is essential due to the constraint on the length of the input document. Hence, keywords are added to the document in order of importance and relevancy until the allowed maximum length of the document is achieved.

Our approach is easily tunable to different languages. In addition, as previously explained, due to the lack of the required information in knowledge bases and also the

complexities of using them, we do not exploit them in our approach. However, we take advantage of web content in order to collect the required data for our approach. As mentioned in Chapter 3, the web contains a great amount of data in various domains and languages that is updated regularly. It also contains both generic and specific terms, which makes this content usable for our approach. Nevertheless, the amount of web content is huge and may contain irrelevant or unreliable information. Analyzing this amount of data is complex. In addition, our enrichment application must make a balance between the amount of the reduced semantic gap and the length of the enriched document. It is also obvious that the recommended information to the user must be as relevant and as reliable as possible. In order to obtain such data, we make use of a search engine as a tool for filtering the web content and returning the most informative and the most relevant parts of data. Therefore, our knowledge-poor approach highly depends on search engines and their top ranked results in the search engine result page (SERP). Changes in the content of SERP could affect the enrichment information recommended by our approach. We note that the web content could be noisy. In Chapter 7, we discuss that our approach is robust to this problem.

Search engines use various criteria in their ranking algorithms. These criteria are not only content-based, but are also based on the link structure of web pages, *i.e.* their inner, incoming and outgoing links. Although linked-based criteria are important in ranking web pages, content-based criteria have the main contribution in this procedure. The details on the ranking algorithms of search engines is, however, out of scope of this thesis. Here, we only take advantage of the assumption that the content of the highly ranked pages in SERP can be considered as informative and relevant source of information. It should be mentioned that this search engine-based approach must be optimized in the number of requests sent to search engines.

The search engine result page typically contains two types of results: organic results and paid results. Organic results list web pages that appear as a result of the search engines' algorithms, whereas paid results show web pages which have been paid to be displayed in the result page as advertisements. Figure 4.2 illustrates an example of the organic and the paid results returned for the query "assurance auto". In our approach, we exploit only the organic results as they contain more informative content and are expected to be less biased than the paid results.

We rely on Google search engine as the most widely-used search engine in 2017[1], although any search engine can be used for collecting the required data. As a future work, we can study the effect of different search engines on the performance of our proposed approach.

In the following, the four steps of our approach are explained but the main steps of the approach, *i.e.* keyword extraction and topic detection, are presented in details in Chapter 5 and 7, respectively.

[1] https://www.netmarketshare.com/search-engine-market-share.aspx?qprid=4&qpcustomd=0

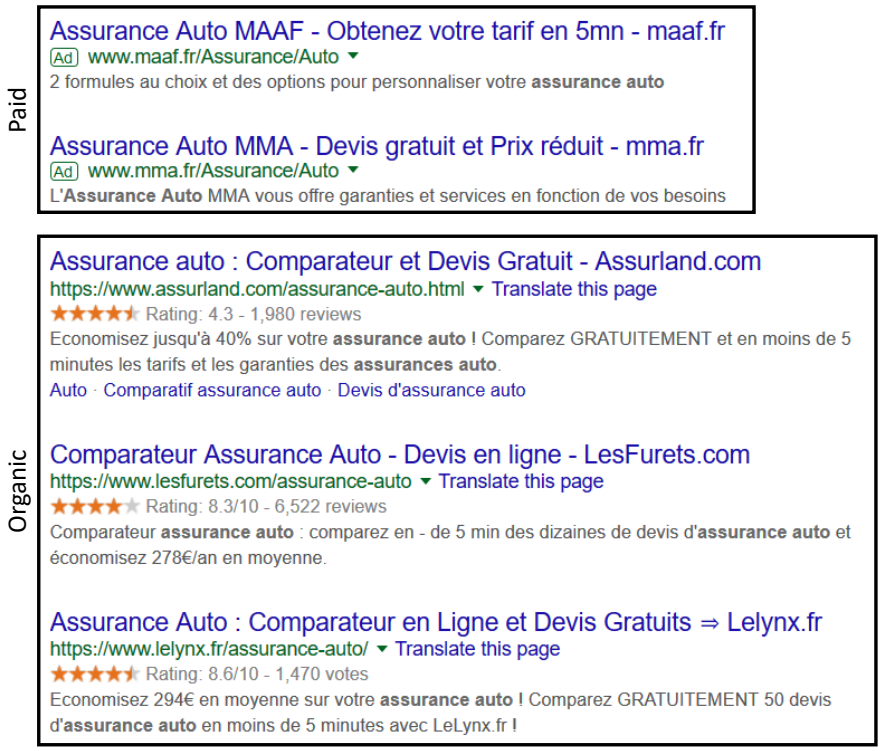


Figure 4.2: Organic and paid results returned by Google for "assurance auto" query

4.2.1 Enrichment collection generation

In the enrichment collection generation step, we generate source of information for enriching an input document with respect to a target domain, which specifies the enrichment point of view. By "collection", we mean a set of representative documents in the domain. Due to the complexity of analyzing a tremendous number of documents in a domain, a representative set of documents should be selected in order to target the most informative and the most relevant documents in the domain. This set can be given as an input data by the user in case that a specific set of documents needs to be considered as source of information. However, this collection is usually not available *a priori* and we found it very demanding for users to generate it manually. Hence, we propose an automatic approach of enrichment collection generation.

We recall that the inputs of our enrichment application are a document, which needs to be enriched, and a keyword, which represents the enrichment point of view, *i.e.* the target domain. In the enrichment collection generation step, we make use of the input keyword to generate a collection of documents, which represents the domain of study.

Generally, the choice of the collection generation approach depends on the application and the type of the input document. As an example, in case of having scientific papers, a "key" word or phrase in a domain can be queried in some databases in order to retrieve papers which discuss that domain. Examples of such databases are Academic Search, PubMed, ArXiv, etc. The returned documents then generate the enrichment collection. In

this thesis, we specifically focus on enriching web pages. As a result, the proposed approach for generating the enrichment collection is also specific to this type of documents.

In order to find a set of web pages related to a certain point of view, we make use of the information provided by search engines, since they effectively return a ranked list of web pages. The highly ranked pages in the search engine result page (SERP) are among the most relevant pages to a query and so can be used as representative documents in the domain of the query. As mentioned before, in this work, we make use of Google search engine for this purpose.

In our approach, the input keyword is firstly queried in a search engine and a desired number of web pages[2] in the result page are collected and considered as the enrichment collection. Although the web pages in the generated collection are ranked based on their relevance to the query, to simplify the problem, we treat them equally and assume them to have the same level of relevancy. Besides simplification, the motivation behind making this assumption is that the highly ranked pages mainly have rich content and their ranking does not only rely on their content. Since in this thesis we are only interested in the content of web pages and not other criteria, such as their link structure, we do not take the different order of the top pages into consideration.

As mentioned before, we only exploit organic results of SERP to focus on more informative and less biased content. The documents collected in this step have the following properties:

- Depending on the input query, the returned pages can be informative, commercial, or a combination of both. By informative pages, we mean web pages which merely give information about the query without aiming at performing commercial transactions, such as selling a product. Wikipedia articles are examples of informative pages. This type of web pages are more likely to be returned for more generic keywords. In contrast, commercial pages aim at selling a product or a service. One of the main characteristics of the commercial pages is that they contain "Add to basket" or similar terms, which indicate a commercial transaction. This type of web pages are expected to be returned for both generic and specific keywords.

- The returned pages must contain valuable information for our enrichment application. In this work, we did not find social media sites and dictionary pages informative enough. Hence, these pages are filtered out in the enrichment collection generation step.

- The returned pages are expected to have the same format: we filter out all pages which do not have any corresponding HTML source code. Examples are PDF files, Word files, etc. As explained in Chapter 2, the choice of the keyword extraction approach highly depends on the format of the studied document. Hence, having

[2]The number of documents in the enrichment collection can be set arbitrary depending on the target point of view. However, the complexity issues should be taken into consideration while setting this number. In this thesis, we set the number of documents to 20 and we use this setting throughout the work.

more than one format of documents, it is likely that different approaches of keyword extraction are required in order to analyze each format individually. In order to simplify the analysis of documents in the enrichment process, we collect the same format of documents in this step so as to use a single approach of keyword extraction. It should be however noted that our enrichment approach is robust to the case where various formats of documents exist in the collection. In this case, the only concern is to adapt the keyword extraction step of the approach to different formats of documents.

- The returned pages must have the same language as the input document, which is aimed to be enriched. Therefore, in this step, all pages which have other languages than the input document are filtered out.

- We need diverse and heterogeneous documents to have a rich enrichment collection and to finally recommend a rich set of keywords to users. To achieve this, in the search result page, we do not select more than two pages from the same website in order to not be biased by the content of one specific website.

We should also point out that in case of having an ambiguous keyword as a query, the returned results by search engines contain pages from different topics. Hence, the collection will not be specific to a single point of view. As an example, two different meanings of "poêle" can be seen in the result page of this query (Figure 4.3). This ambiguity is managed in our proposed topic detection approach (see Chapter 7).

4.2.2 Keyword extraction

One important step of our approach is to extract the main information of the input document and the associated collection. As explained in Chapter 2, manual analysis of documents is impossible. Hence, an effective approach is required to analyze the documents and to extract their main information as a set of "keywords". Depending on the target application, keywords of various length are extracted. In our approach, we are interested in both single and multi-token keywords. The motivation behind extracting multi-token keywords is to capture the association between the words and to extract more information from documents. However, as will be explained in Chapter 5, we limit the length of multi-token keywords to avoid getting over specific ones.

In Chapter 2, we detailed the different approaches of keyword extraction proposed for various types of documents and applications. Considering the extraction features and methodologies presented in that section, we propose a keyword extraction approach, which specifically aims at extracting keywords from web pages. The proposed approach is unsupervised and so does not suffer from the complexities of generating a training set. We also do not want to rely on a corpus of side documents to extract keywords from a specific document. Hence, our approach is document-centered and not corpus-centered. In addition, no knowledge-base is exploited so that our method be as generic and as domain-

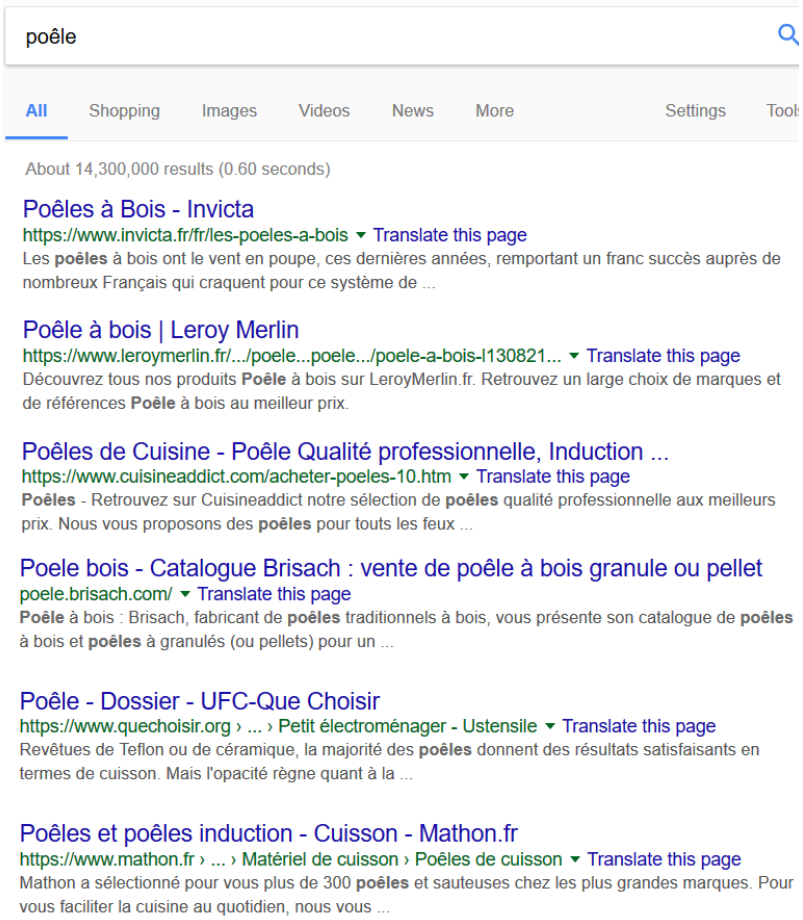


Figure 4.3: Example of an ambiguous query and its multi-topic results

independent as possible. Our approach was initially proposed for French but it is easily tunable to other languages. More specifically, we tried languages which are cognate with French and are typologically similar to it. As a result, in addition to French, the approach has been implemented on English, Spanish, Portuguese, Dutch, German, Estonian, Bulgarian, Russian, Polish, Italian, and Romanian. It has been also tested by native speakers on the first four languages. However, in this thesis, we formally evaluate the approach only on French language. We note that an analysis on the categories of languages on which we can perform our keyword extraction remains to be done.

In order to extract keywords from a given web page, first the content of the page is extracted and uninformative parts of the content are removed. The extracted content is then analyzed for extracting its main information. We recall that there are two main strategies for extracting keywords. The first strategy is to extract an initial set of phrases and to rank them according to different criteria. The highly ranked phrases are then considered as the extracted keywords of the studied document. The second strategy, however, starts with extracting a set of important single words and then expands them to multi-token keywords according to their co-occurrence in the document. Our approach follows the second strategy. After extracting the content of a page, all the single words of the page

are returned. These words are then scored according to their importance in the page. The highly ranked words are then returned as the most important words. Eventually, based on the co-occurrence between the returned words, multi-token keywords are generated and considered as keywords of the studied page. The keywords are scored according to the score of their constituent words. More details on the keyword extraction approach are presented in Chapter 5.

4.2.3 Topic detection

After generating a collection of documents for the enrichment process and extracting their main information as a set of keywords, we cluster those keywords into topics. This is the topic detection step. We recall that by "topic", we mean a set of semantically related keywords. For our application, we are not only interested in detecting semantically similar keywords, but we also aim at detecting semantically related ones. We try to capture all synonymy, meronymy and hyponymy relations in the topic detection step. Our approach is also able to detect polysemies in case of having them in the collection of keywords. As keyword extraction, our topic detection approach is knowledge-poor and so domain-independent. It is also easily tunable to different languages. The approach has been already implemented and tested on French and English but the formal evaluation of the approach has been done on the former language. We believe that unlike keyword extraction, the topic detection approach is not specific to web pages and can be applied on any type of documents once the set of their corresponding keywords is provided as an input data. However, this property of the approach needs to be experimented and evaluated in the future.

The topic detection step consists of two phases: coarse-grained topic detection and fine-grained topic detection. Both of these phases make use of graph-based approaches, which model the semantic relatedness and the semantic similarity between keywords of the collection. However, the details of the graph-based approaches differ in the two phases and are explained in Chapter 7. The motivation behind using graphs is that they explicitly model the relation between keywords. Various graph analysis approaches can be exploited in order to retrieve different kinds of information from the graphs. It should be noted that by using a graph-based model, the problem of topic detection is regarded as a community detection problem, where the goal is to detect the communities within the graph. Each community consists of a set of highly connected keywords, which is referred to as a *coarse-grained topic* in this thesis. Keywords in coarse-grained topics are "semantically related". We use overlapping community detection algorithms because of the problem of ambiguity in the collection.

Applying the coarse-grained topic detection phase is essential in order to distinguish the main topics of the collection, to disambiguate possible polysemies and to identify the topic(s) which are related to the domain of interest. However, these topics are too generic to be directly recommended to the user and are mostly large in terms of their constituent keywords. For the enrichment process, a more structured recommendation

is required. Hence, we further divide each coarse-grained topic into sub-topics, called *fine-grained topics* in this thesis. Having these topics, the recommended information is well-organized and can be exploited in a more effective way by the user. After detecting the fine-grained topics, keywords of each topic are ranked based on their importance and relevancy to the topic. The ranking is performed according to the information obtained from the generated graph and also the score of the keywords, assigned in the keyword extraction step. Recommending a ranked list of keywords is to satisfy the constraint on the length of the input document. Having the ranked list, the user adds the keywords in order of importance and relevancy until the allowed maximum length of the document is obtained.

In Chapter 7 we explain in more details the topic detection step, which is the main contribution of this thesis.

4.2.4 Filtering

Before passing the fine-grained topics to the user, we perform a minor analysis in order to avoid recommending keywords which already exist in the content of the input document. This analysis is performed in the filtering step. Having a web page as the input document, the proposed keywords are compared with the core content of the page. However, we did not find the exact match between the keywords and the content of the page effective enough. Since keywords may consist of more than one token, finding the exact match is a too strict condition. In other words, it is likely that a keyword remains implicit in the content of a document and no exact match is found between the document and the keyword. As an example, the page content in Figure 4.4 discusses "poêle anti adhésive", even if it is not explicitly mentioned in the text. Adding this keyword to the page would wrongly generate duplicate content.

To overcome this issue, we assume that if at least one sentence of the input document contains all the tokens of the target keyword, the keyword should not be recommended to the user. According to our assumption, the existing tokens in the sentence should not be necessarily adjacent and in the same order.

At the end of the filtering step, the remaining keywords in the fine-grained topics are recommended to the user for the enrichment purpose. In the rest of this thesis, we do not come back to details of this step and we assume that after topic detection step, we have the final lists to recommend to the user.

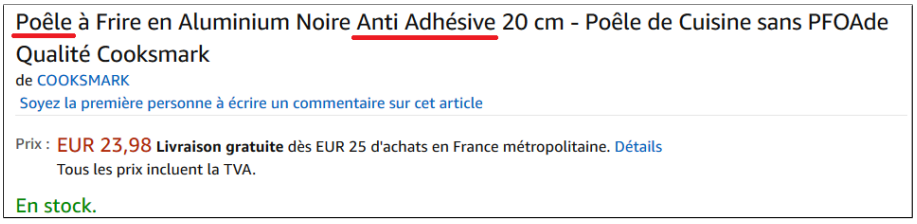


Figure 4.4: Content which implicitly contains the keyword "poêle anti adhésive"

Part IV

Keyword extraction

Chapter 5

Keyword Extraction Methodology

Contents

Keyword extraction is one of the main steps of our enrichment approach, where important lexical units of documents are extracted as summaries over them. The set of documents analyzed in this step consists of an *input document*, which the user wants to enrich, and additional documents collected for the enrichment of the input one. The user of our approach is a person who performs the enrichment procedure on the input document and specifies the enrichment's point of view by entering a keyword as an input (see Figure 4.1). As explained in Chapter 2, manual analysis of documents is very laborious and time-consuming. Depending on the number of available documents, it can be even impossible. Hence, an automatic approach of extraction is required for their analysis. In Chapter 2, we introduced different approaches of keyword extraction, proposed for various types of documents and applications. In this thesis, we specifically focus on enriching the content of web pages. Hence, our keyword extraction approach is proposed for extracting keywords from web pages[1]. In this work, a "web page" is a document which has a corresponding HTML source code. Other types of documents on the web which do not have this source code, such as PDFs, PPTs, etc., are not relevant for our enrichment application and so are discarded in our approach.

Our keyword extraction must be domain-independent to be applicable on web pages of different domains. To meet this constraint, we do not exploit any external knowledge base and our extraction features can be applied on a page regardless of its domain. Since exploiting domain-dependent information can increase the effectiveness of the keyword extraction approach, we use it as an optional information when it is available. We nevertheless consider that our basic keyword extraction approach is domain-independent. This approach is also easily tunable to different languages.

Unlike many approaches which extract keywords of a document using the information obtained from other relevant documents, our approach does not require this information.

[1]This chapter is based on a patent entitled "Procédé d'extraction de mots clés, dispositif et programme d'ordinateur correspondant", written by Nazanin Firoozeh, Fabrice Alizon and Adeline Nazarenko that has been submitted on 13 July 2015.

In other words, we focus on a document-centered approach rather than a corpus-centered one as the collection of relevant documents is not always available and generating it might be demanding for users. Moreover, we propose an unsupervised approach of extraction, since generating a training set which can be used for different domains of web pages is challenging (see Chapter 2) and a supervised approach may not return robust results.

In our work, we extract information of a document as both single and multi-token keywords. The motivation behind using multi-token keywords is to consider the strong association between single words and to extract more informative and semantically autonomous lexical units. Among the different properties of keywords, which were listed in Chapter 2, we are mainly interested in *representativity* and *specificity* of the extracted keywords and their *cohesiveness* in case of having multi-token keywords. Hence, our exploited extraction features aim at targeting these three properties that we found important for our enrichment application.

Multi-token keywords, however, can be of various lengths in terms of the number of tokens. In order to specify the allowed length of keywords, we make use of queries of search engines as indicators of the most common length of a keyword. Here, our assumption is that queries have the same characteristics as our desired keywords and so can give information about their common length. Queries are made by people who use a search engine for different purposes and are stored as the query log of that search engine. It is also obvious that expert people make better queries in search engines comparing to non-expert ones. Considering these points, we make use of the analysis of Fang et al. (2011), which was performed on search queries made by expert people. These queries have been obtained from INDURE[2]. According to their result, shown in Figure 5.1, people mostly search for queries with length of two and it is rare for an expert to search for queries with more than five tokens. According to these findings, we assume that effective keywords contain up to five tokens, without considering stop words[3]. Since there is no single list of stop words that can be universally used by different natural language processing tools, we generate our own list depending on the requirements of our application.

We recall that there are two main strategies when extracting keywords from documents. In *synthetic* strategy, the goal is to extract an initial set of phrases from a document and then to rank them in order of their importance to get the final set of important phrases, considered as keywords of the document. The *analytic* strategy, however, starts with a set of single words of a document and expands them to possibly compound words in order to have a representative set of keywords. Synthetic strategy is more linguistic-based, whereas analytic strategy is language-poor and can be easily adapted to unstructured documents and new languages. So it is a more robust strategy compared to the synthetic one. Our initial tests showed that the writing of web pages is not very standard. Hence, we did not

[2]https://www.indure.org/

[3]According to Lo et al. (2005), stop words are defined as "words in a document that are frequently occurring but meaningless in terms of Information Retrieval (IR)". Examples of English stop words are *a*, *an*, *the*, *by*, etc.

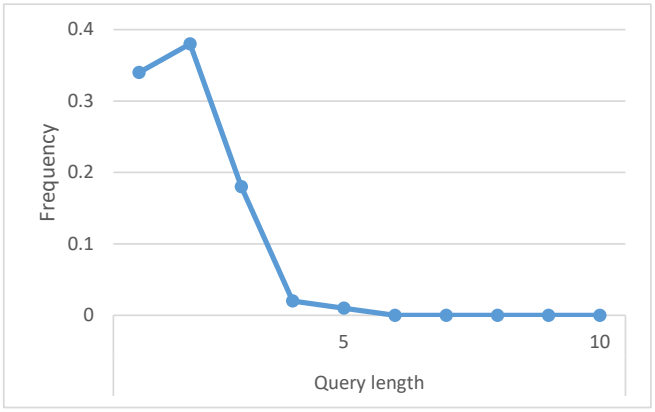


Figure 5.1: Distribution over the number of tokens in search queries (Fang et al., 2011)

find linguistic-based tools, such as Named Entity Recognition tools, to perform well. Due to these reasons, we follow the second strategy in our keyword extraction approach.

Our goal is to generate well-formed keywords out of the main words of a web page. For this purpose, we focus on both precision and recall measures. Although both of these measures are important, since recommending ill-formed keywords to users is not acceptable, precision is more important in our application.

It is mainly assumed that the number of the extracted keywords is a function of the length of the studied document and more keywords are extracted from longer documents. In this work, we discuss that this is not always a true assumption and representativity of words within the studied document is also an important factor in determining the number of the extracted keywords: not many keywords are expected to be extracted from a long document with insignificant words and vice versa.

In our approach, the extracted keywords are ranked in order of their importance in documents. In some applications, a fixed number of keywords is required. Using the ranks of keywords, one can select the most important keywords for such applications. As will be discussed in Chapter 7, the length of the input document needs to be controlled in the enrichment application and quite often not all the recommended keywords can be added to the document. Hence, we associate a weight to keywords to take their importance into account. In the following steps of the enrichment approach, the weights are updated based on various criteria (see Section 7.2) and finally the keywords are recommended in order of their weights.

Figure 5.2 illustrates the overall framework of our keyword extraction approach. It consists of three main steps: *text analysis, top words selection*, and *keyword generation*. The input web page can be either the page which is supposed to be enriched or the ones collected in the enrichment collection generation step (see 4.2.1). In the first step, the input web page is processed and its informative content is extracted. Further processing is then performed on the extracted content to return the candidate words of the page, which have the minimum expected informativeness. The second step analyzes the candidate words

and scores them according to their importance in the page. The top ranked words are then selected and passed to the keyword generation step, where keywords of the page are generated out of the top words. The number of the generated keywords depends on both the number of the selected top words and the way they are associated in the content of the studied web page. In the following, we explain each step of our approach in more details.

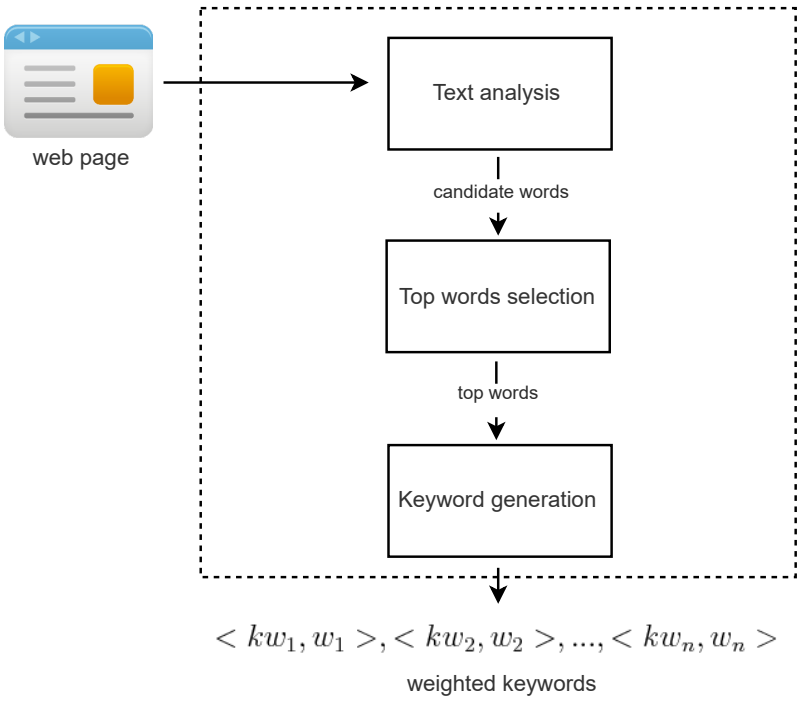


Figure 5.2: Overall framework of the keyword extraction approach

We note that although the proposed keyword extraction approach is used as one of the main steps of our enrichment approach, it can be used separately in other applications, where representative lexical units of web pages need to be extracted.

5.1 Text analysis

In the first step, we aim at parsing the HTML source code of the input web page in order to get its content for further processing. Different parts of the page are not equally informative. Some parts are mainly related to the template of the page, which does not bring any particular information about its content. As an example, a heading, with phrases like “Add to cart” and “Sign up”, gives no information about the page. For our enrichment application, for example, information related to the price of a product is not relevant for enriching a specific document. Moreover, some parts of the page contain considerable information but this information is related to the whole website and not a specific page. Since our goal is to extract keywords which are specific to the content of the studied page, we eliminate those parts of the page in order to only focus on the page-specific content. A

typical example of these contents are menus, which appear in many pages of the website and summarize their information. Hence, unlike some of the keyword extraction approaches in the state of the art, we exclude the uninformative and the generic parts of the web page in order to increase the informativeness and the specificity of the extracted keywords.

Figure 5.3 gives an example of a web page along with its corresponding informative and uninformative contents. We refer to the collection of all the desired contents of a web page as its *core content*. After extracting the core content of a page, a pre-processing step is performed in order to make the content ready for further analysis, where we extract the sentences of the page and its *candidate words*. Candidate words are the ones which are informative enough to be considered as constituent units of our final keywords. In the following, we explain our pre-processing and detail the two kinds of data that we obtain in the text analysis step.

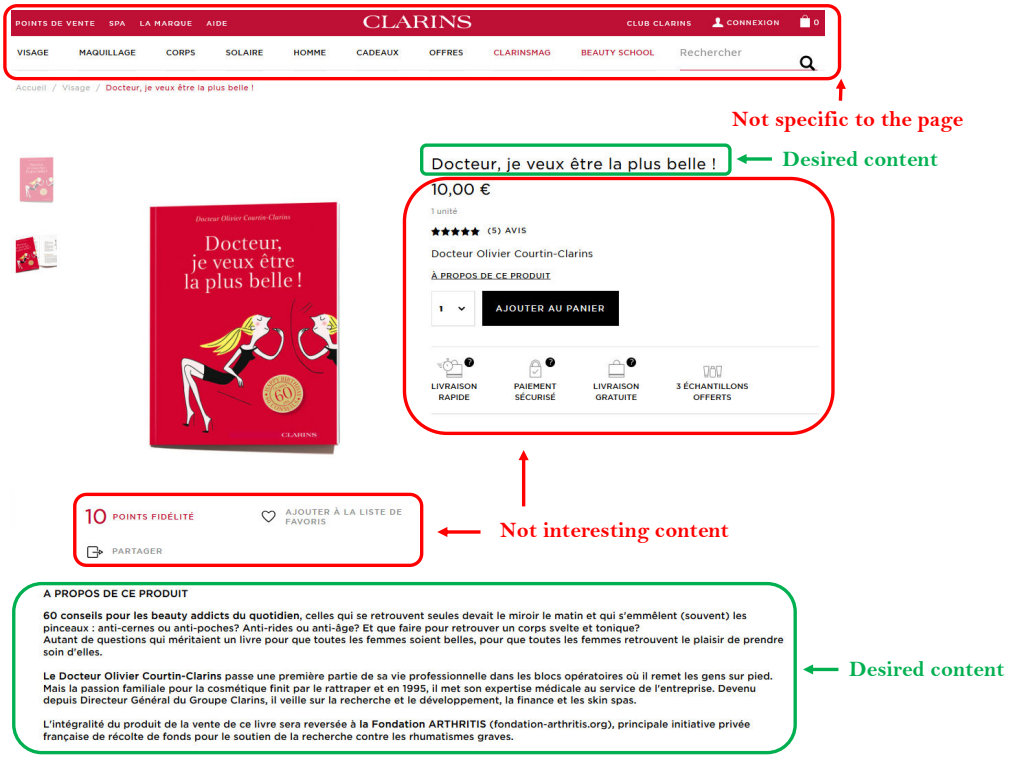


Figure 5.3: Example of the core content (desired contents) of a web page

Pre-processing. The goal of the pre-processing step is to make the core content of the studied page ready for further analysis. As explained above, we remove the uninformative and generic parts of the web page content. Here, we also discard the pieces of text that are not informative enough for our enrichment application. More specifically, we remove three types of information: emails, URLs, and terms which appear frequently in commercial web pages. We refer to the latter elements as *e-commerce expressions*. Examples are "customer review", "free delivery", "means of payment", etc. Of course, this processing is very depending on the type of application that is targeted.

In addition to removing uninformative content, we lemmatize the core content in order to reduce inflectional forms of words to their common base forms. Using lemmatized forms increases the accuracy of matching words throughout the approach. In our approach, we exploit *TreeTagger*[4] for the lemmatization purpose. Once lemmatized, the informative core content is further analyzed to segment its sentences and to extract its candidate words.

Sentence segmentation. The goal of sentence segmentation is to avoid crossing sentences or blocks boundaries while generating keywords in the keyword generation step. In other words, using segmented sentences, we eliminate trivial errors and reduce the running time of the approach. To segment the core content into a list of sentences, we make use of delimiters that we empirically found effective as indicators of the sentences boundaries. Those delimiters may vary from one language to the other and depend on the target application. Table 5.1 lists the delimiters that we use for sentence segmentation.

Table 5.1: List of the delimiters used for sentence segmentation

.	:	,	!	...	. . .	?	;	(	)	[	]	{	}	<	>	«	»	≫	≪

We consider "." and "," as delimiters only if they are followed by a space. As an example, "F.A.Q" has no delimiter and is not split into different sentences. This rule, however, is not always valid due to the diversity in word and abbreviation spelling. Specifically, in honorific and some specific formats of numbers, considering ". " and ", " as delimiters may wrongly split a term into different sentences. Examples are "10. 000", "10, 000", and "Mr. Smith". In order to manage this diversity, before the segmentation, we apply some patterns to normalize the spelling of certain types of words. As an example, "10. 000", "10, 000", and "Mr. Smith" are respectively transformed into "10.000", "10,000", and "Mr Smith". Consequently, no delimiter splits them wrongly.

Extracting candidate words. By extracting candidate words of a web page, we eliminate words which have a little chance of being a constitutive base of our desired keywords. These uninteresting words are mainly the ones which contain no or little information. Hence, removing them leads to generating more informative keywords. To specify the informativeness of words, we make use of their *morpho-syntactic feature*. More specifically, we use the part-of-speech (POS) tags of all words of the page to identify the candidate words. We empirically found *common* and *proper nouns, verbs, adjectives, negative adverbs, numbers*, and *abbreviations* to be informative for our application. In addition to filtering out the uninformative words, this POS-filtering step reduces the number of the candidate words and lowers the complexity of the keyword generation step. In our approach, stop words are also not interesting to be considered as candidate words of a page and so are filtered out. However, as will be explained in the keyword generation step, some stop words, such as prepositions, are used for merging words and generating keywords. The final set of candidate words is later used in the top words selection step, where the

[4]http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

words are ranked in order of their importance and the most important ones are selected for the keyword generation step.

An example of the explained steps is illustrated in Figure 5.4. In this example, the original content is lemmatized in a pre-processing step. The candidate words are then identified in the lemmatized content, shown as underlined words in the figure. Eventually, sentences of the text are segmented and returned.

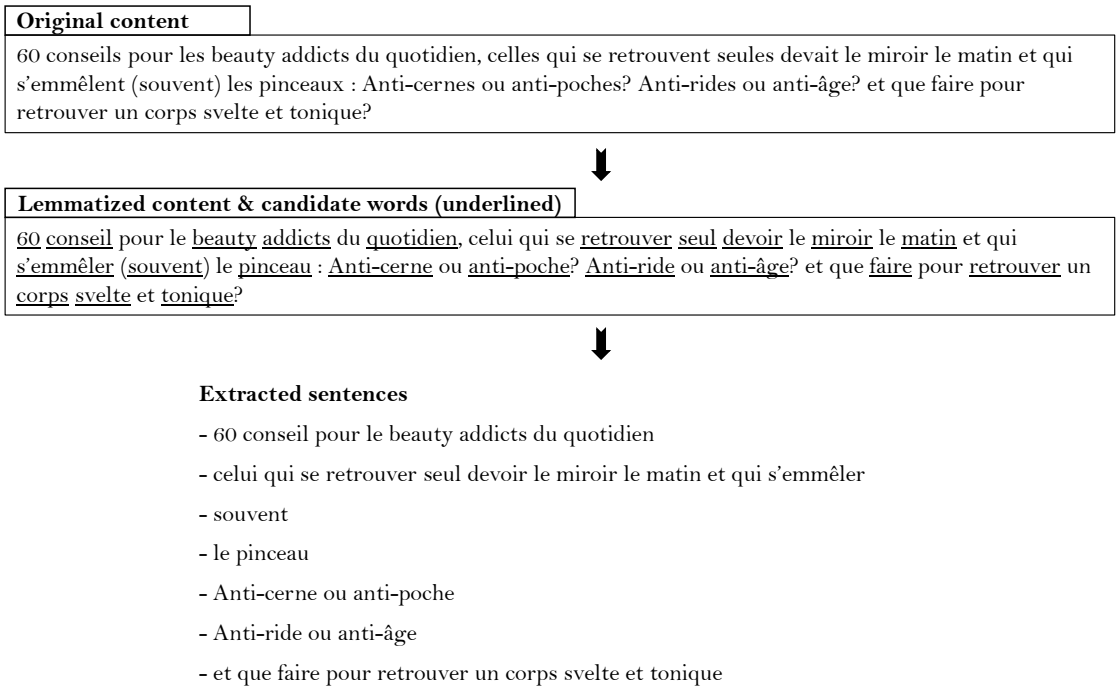


Figure 5.4: Example of the extracted sentences and the identified candidate words

5.2 Top words selection

Having the list of the morpho-syntactically filtered words, *i.e.* the candidate words, the next step consists in determining their importance in the studied web page. We note that a word can have different levels of importance in different documents depending on their topics. Once the importance of words is determined, the words with high importance are returned as top words of the page. We focus on top words to satisfy the representativity property of the final keywords. Stated differently, using top words, we avoid generating keywords which are not good descriptors of the studied web page. In addition, by taking only the top words into account, the number of words that are passed to the keyword generation step decreases considerably and this reduces the complexity of our extraction approach. In Section 6.1, we show that the optimal number of the selected top words depends on the representativity and the importance of the words within the studied page.

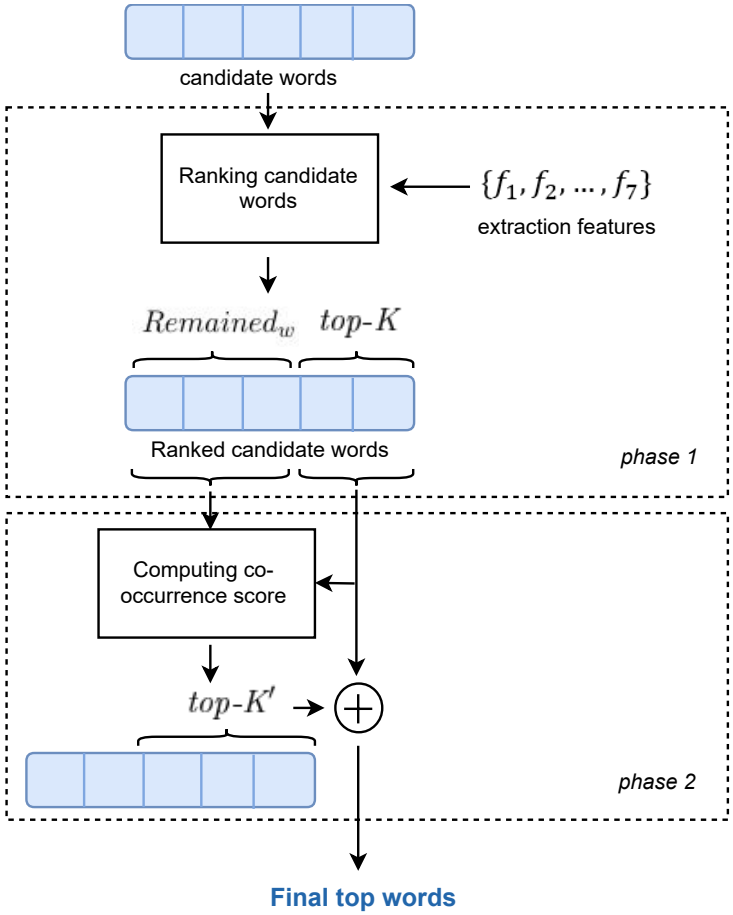


Figure 5.5: Overall framework of top words selection step. Phase 1 and phase 2 are respectively related to "selecting an initial list of top words" and "expanding the list of top words".

Top words of a web page are extracted in two phases: In the first phase, an initial list of important words are selected using some extraction features. The second phase then expands the initial list using the notion of co-occurrence between the selected words and the remaining ones. Eventually, the collection of all the words selected in the two phases is considered as the top words of the page.

The two phases are respectively called *selecting an initial list of top words* and *expanding the list of top words*. An overview of the top words selection step is shown in Figure 5.5. In the following, each phase of extraction is explained in more details.

5.2.1 Selecting an initial list of top words

In this phase, all the candidate words of a page are ranked according to some extraction features that are explained in the following. The *top-K* words in the ranked list are then considered as the initial top words of the page.

Extraction features

To determine the importance of the candidate words, we make use of three types of extraction features: *statistical, informational,* and *resource-based* (see Section 2.5). Each type of features brings particular information for determining the importance of the candidate words and satisfies a specific property of the target keywords. In the following, we detail the way that these features are used in our approach. In the next step, a feature vector is assigned to each candidate word, where each element is related to one exploited feature. The overall importance of the words is then determined by aggregating all the features in the vector.

Statistical features. A traditional feature used in most of the keyword extraction approaches is the frequency of words within the studied document. The assumption behind using this feature is that the more frequent a word is in a document, the more important it is in that document. We also make use of this assumption and exploit the *normalized frequency* as one of the features for determining the importance of the candidate words. To normalize the frequency values, we divide them by the frequency of the most frequent word in the page. This feature is used to mainly satisfy the representativity property of the final keywords, where the more frequent keywords can be considered as better descriptors of the studied page.

To satisfy the specificity property of keywords, we tried to introduce the TF.IDF measure in our approach but it did not bring additional information about words. This can be explained by the fact that we analyze only the core content of web pages and the generic content, which brings information for TF.IDF, is removed *a priori*. We suppose that TF.IDF can be used instead of normalized frequency, if we were not extracting the core content of the page. We, however, found extracting the core content to be less complex than calculating TF.IDF. In addition, a corpus of relevant documents for computing this measure may not be always available. As a result, to reduce the complexity and to have a document-centered approach, we do not exploit this feature in keyword extraction.

Informational features. Representativity of the final keywords is also satisfied by exploiting informational features in selecting the most important words of the studied page. Informative areas within a document are considered as indicators of word importance. Depending on the type of the document these areas may vary. To detect the informative areas within a web page, we make use of the basic search engine optimization techniques (Google, 2010). According to these techniques, some parts of a web page are important in the ranking algorithm of search engines. Hence, they are more likely to contain more informative and more page-specific content. In this work, we focus on the following areas as the informative parts of a web page:

- *Title*: the title of a web page is the content found in the *<title></title>* tag of the page's HTML code. Visiting the web page, this content is displayed at the top of the browser.

- *Meta description*: this tag gives a summary of what the page is about. Meta description content is found in the "content" attribute of the following HTML tag:

  <meta name="description" content="Summary of the page...">

- *Image alt*: in our approach we only process textual content and not other types, such as images, which may contain considerable information in web pages. In order to make use of the information that images bring, we exploit their corresponding *alt* attribute. This attribute basically provides alternative information for an image if a user cannot view it for some reasons. More specifically, we make use of the *alt* content in the following HTML tag:

  <img alt="The image description">

- *URL*: in search engine optimization, web pages are recommended to have descriptive URLs with relevant words to the page content. Considering this guideline, we make use of URLs as an informational feature in our approach. It should be however noted that we did not find the hostname and clearly the protocol of URLs to be informative for our application. Hence, other parts of URLs are considered as source of information. Figure 5.6 shows an example of the informative part of a URL in our application.

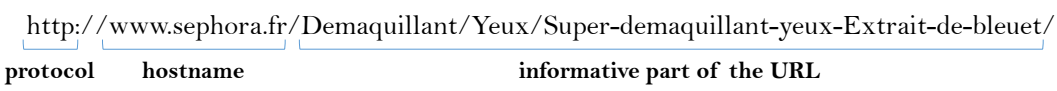


Figure 5.6: Informative part of an example URL

In addition to the informative areas, we also rely on the assumption that the more informative words tend to appear earlier in the content. Unlike some works which use the average position of a word for this purpose, we found the position of its first occurrence to be a better indicator of its informativeness. We refer to this informational feature as the *position* feature and compute it using Equation 5.1, where $position(w_i, p_j)$ is the position feature of word w_i in page p_j, $FirstOcc(w_i, p_j)$ gives the ordinal number of the first sentence of p_j that contains w_i, and $|sentences(p_j)|$ is a normalization factor, which indicates the total number of the sentences in p_j.

$$position(w_i, p_j) = 1 - \frac{FirstOcc(w_i, p_j)}{|sentences(p_j)|} \tag{5.1}$$

Resource-based features. In addition to the explained domain-independent features, we also exploit a resource-based feature in order to study the effect of domain-dependent

information on our extraction approach. We have been inspired by Yih et al. (2006), who found query logs of search engines as one of the most important sources of information in extracting advertising keywords. However, we use query logs statistics in a different way. While Yih et al. (2006) use query logs to get information about candidate keywords, we use them at the word level in order to determine the importance of words. Our assumption is that the more frequent a word appears in a query log, the more relevant it is to the subject of the website. This feature mainly satisfies the specificity of the final keywords. We empirically found out that this domain-dependent information improves the effectiveness of the approach. Nevertheless, query logs are not always available. In addition, due to search engines policies, the amount of available information is decreasing considerably as time goes by. Our approach is robust to this limitation, since the extracted keywords are effective enough for the enrichment application, even if we do not exploit query log information. Therefore, this feature is considered as an optional feature in our approach.

Our features are comparable with the ones exploited by Yih et al. (2006). However compared to this work, we use a fewer number of features and this decreases the complexity of the approach. In addition, the general strategy differs in the two approaches. While we aim at detecting the important words of the page to further generate the keywords out of them, they follow the synthetic strategy. Hence, in their work, the features are used to determine the importance of candidate keywords rather than single words. Moreover, they have a supervised approach, where the features are used to train a classifier. Therefore, their approach requires the generation of training data.

Ranking candidate words

As a summary over our extraction features, the importance of each candidate word is determined using seven features: normalized frequency (NF), occurrence in title (T), occurrence in Meta description (M), occurrence in image alt (I), occurrence in URL (U), position of the first occurrence (P) and query log statistics (Q), which is used in case of availability. These features belong to three categories of extraction features presented in Section 2.5: statistical, informational and resource-based. Considering these features, we assign a feature vector FV to each candidate word w_i, which is represented as: $FV(w_i) = \{NF(w_i), T(w_i), M(w_i), I(w_i), U(w_i), P(w_i), Q(w_i)\}$.

In this vector, NF, P, and Q features are scalar, whereas the other features are modeled as boolean values. All the features are language independent and all except Q, which is an optional feature, are domain independent and do not require any corpus or external resource. Having these features, the associated score to each word is computed based on Equation 5.2, where $feature_j$ corresponds to the j-th element of the feature vector and n is the total number of the features, *i.e.* 7.

$$Score(w_i) = \sum_{j=1}^{n} feature_j(w_i)^2 \tag{5.2}$$

We initially set the value of K to 10 and select the 10 words of the candidate list with the highest scores. This number has been determined empirically and we also found it to be inline with the choices of Vidal et al. (2012) and Wan and Xiao (2008b), who consider the top-10 keywords of a web page as the most relevant ones. We, however, increase this number as long as the difference between the score of the next word to analyze and the 10th selected word is lower than a pre-defined threshold value.

Focusing on scores of words rather than the length of pages shows that in our approach, representativity and importance of words within the studied page are important factors in determining the number of the selected words. Our studies on different web pages show that long pages are not necessarily informative. On the other hand, a considerable amount of information may be put in pages in a very concise way. Due to these findings, we mainly focus on the content relevance rather than the document length but we also experiment the impact of the length in our work. In Section 6.1, we show that the length of a page is not an effective factor for determining the number of the selected words. The number of the extracted keywords by our approach depends on both the number of the selected words and the way they are associated in the content of the page.

5.2.2 Expanding the list of top words

In the second phase of the top words selection step, inspired by Matsuo and Ishizuka (2004), we assume that if a word appears frequently with a subset of important words, it is also likely to be important. To be clearer, the set of the remaining candidate words after the first phase of extraction is named $Remained_w$. According to the number of co-occurrences between $Remained_w$ and *top-K* words, we select a second set of words from the studied page. To achieve this, we first create a matrix of co-occurrence in order to store the co-occurrence statistics between $Remained_w$ and *top-K* words. We consider two words to be co-occurrent if they appear within the same sentence. Element $w_i w_j$ of the matrix shows the number of times that words w_i and w_j appear together in sentences of the studied page.

A co-occurrence score is assigned to each $Remained_w$ word based on the number of co-occurrences between that word and all the *top-K* words. In this work, we tried three existing measures for computing the co-occurrence score and proposed a new one that is explained in the following. In Equations 5.3 to 5.7, w is a word from the $Remained_w$ set for which the co-occurrence score is calculated and G is the set of the *top-K* words, each of which is represented as g. $freq(w, g)$ shows the number of times that w and g co-occur within the same sentence.

- X^2*-measure:* used by Matsuo and Ishizuka (2004), X^2-measure calculates "the degree of bias of the co-occurrence distribution" using Equation 5.3. The degree of bias is

then used as an indicator of word importance. The goal of their work is to extract keywords from a single document using word co-occurrence statistical information.

$$X^2(w) = \sum_{g \in G} \frac{freq(w,g) - n_w p_g}{n_w p_g} \tag{5.3}$$

In Equation 5.3, n_w indicates the total number of times that w co-occurs with words of G and p_g is defined as the unconditional probability of g, calculated using Equation 5.4.

$$p_g = \frac{\#\ Sentences\ where\ w\ and\ g\ co\text{-}occur}{\#\ Sentences\ in\ the\ text} \tag{5.4}$$

- *Improved X^2-measure:* Matsuo and Ishizuka (2004) improved their proposed X^2-measure by taking the length of sentences into consideration. The new measure is similarly computed using Equation 5.3 but with other definitions of p_g and n_w. Here, n_w is the total number of words in the sentences which contain w. p_g is also calculated using Equation 5.5, where S_g is the set of sentences which contain g.

$$p_g = \frac{\sum_{s \in S_g} length(s)}{length(text)} \tag{5.5}$$

- *$Score_{Rose}(w)$:* Rose et al. (2010) proposed a measure for scoring candidate words in a keyword extraction approach. We name this score $Score_{Rose}(w)$ and try it in our approach for computing the co-occurrence score. According to this measure, the co-occurrence score of w is calculated using Equation 5.6, where $deg(w)$ is the total number of words that co-occur with w. Hence, the longer the sentence which contain w is, the higher the value of $deg(w)$ is. $freq(w)$ shows the frequency of w regardless of the number of its co-occurrent words.

$$Score_{Rose}(w) = \frac{deg(w)}{freq(w)} \tag{5.6}$$

  We try this measure to compute the co-occurrence score. The difference is that we have a lower value of $deg(w)$, since in sentences we take only the *top-K* words into account instead of all the words. As an example, having $G = \{A, B, C, D\}$, if w appears 3 times in a content, where it co-occurs twice with A, 3 times with B, and 5 times with D, then the deg and $freq$ values are the following.

  $deg(w) = 2 + 3 + 5 = 10$, $freq(w) = 3$

- *Weight-based Score (WBS):* the weight of words is an important criterion that has not been taken into consideration in any of the previously mentioned measures. Considering this point, we propose a modified version of $Score_{Rose}(w)$, which takes the weights of the studied words into account. We note that *top-K* words do not have the same importance and this is inferred from the scores that were initially calculated using Equation 5.2. Our assumption is that co-occurrence with a more important

word is more significant than co-occurrence with a less important one. Hence, in addition to the co-occurrence frequency, the weights of the co-occurring *top-K* words should be considered in calculating the co-occurrence score. Equation 5.7 shows our proposed score that we refer to as *Weight-based Score* (WBS). In this equation, $score(g)$ is the initial score of words obtained using their corresponding feature vectors (Equation 5.2).

$$WBS(w) = \sum_{g \in G} \frac{freq(w,g) \times score(g)}{freq(w)} \tag{5.7}$$

As an example, let $G = \{A = 0.8, B = 0.4, C = 0.3, D = 0.2\}$, where the values show the scores of the words. Suppose that $Remained_w$ contains two words w_1 and w_2 with frequencies of 2 in the page under analysis. Considering the following table as the co-occurrence statistics between words of $Remained_w$ and G, we compute the values of $WBS(w_1)$ and $WBS(w_2)$ as below:

	A	**B**	**C**	**D**
$\mathbf{w_1}$	0	2	3	1
$\mathbf{w_2}$	2	1	0	0

$WBS(w_1) = \frac{(2\times0.4)+(3\times0.3)+(1\times0.2)}{2} = 0.95$, $WBS(w_2) = \frac{(2\times0.8)+(1\times0.4)}{2} = 1$

In this example, the number of co-occurrent words is higher for w_1, but it gets a lower WBS value compared to w_2, because it is co-occurring with less important words of G.

After calculating the co-occurrence score for each word in $Remained_w$, we select the *top-K′* words with the highest co-occurrence scores. In Chapter 6, we compare the performance of the co-occurrence scores and show that our proposed one outperforms the others for our specific application. In that chapter we also discuss the choice of K' value.

At the end of the top words selection step, we consider all the returned *top-K* and *top-K′* words as the final top words of the studied page. These weighted words are then passed to the keyword generation step, where multi-token keywords are generated out of these words and the final set of keywords is returned.

5.3 Keyword generation

In the keyword generation step, the selected top words are possibly merged with each other to form longer and more specific keywords. Hence, depending on the content of the studied document and the co-occurrence of the top words, some words could be merged to represent more informative and more specific lexical units. In our approach, the extracted information is returned as single and multi-token keywords. We recall that the motivation behind generating multi-token keywords is to take the association between words into account and so to generate more informative keywords. In fact, the extracted words which

are not semantically autonomous may bring wrong information about the studied document. As an example, having a text discussing traffic issues in New York, both "new" and "york" can be extracted as single words. Considering these words individually, one may infer that the text is discussing the "New" traffic issues in "York". Hence, apart from their association, the words may not be good descriptors of the studied text.

More specifically, this step studies the previously extracted top words and returns the final set of keywords. The proposed keyword generation approach exploits the basic statistical and pattern-based methods, explained in Chapter 2. The result of the keyword extraction approach must be robust when applied on different domains. Therefore, we propose a domain-independent approach, which does not exploit any knowledge base for capturing the association between words. In addition, our enrichment application needs to be fast, as it is interacting with users. Hence, complex methods in which time-consuming analysis is performed, do not meet our requirement. Considering these points, we propose an unsupervised approach, which exploits basic methods but returns high quality keywords in a robust way and in a reasonable time. Although the balance between precision and recall values of the extracted keywords needs to be taken into consideration, as will be argued in Chapter 6, we consider that the precision value matters more for our application.

The overall framework of the keyword generation step is presented in Figure 5.7. Keyword generation consists of three steps: *merging adjacent words*, *enlarging keywords to co-occurrent ones*, and *pattern-based filtering*. In this section, we explain each step of the keyword generation in more details.

The main part of our keyword generation step makes use of a statistical method, which generates the candidate keywords. A pattern-based method is then applied as a post-processing step to filter out some specific patterns that we found frequent but not informative for our application. As explained in Chapter 2, the statistical methods exploit statistical features. In our work, we use the co-occurrence-based feature, which captures the association between the input words and possibly expands them to longer keywords. To do this, the window-wise co-occurrence feature is used, where the window has a fixed length which is determined empirically. The co-occurrence measure in our approach is scalar rather than binary, since the frequency of co-occurrence is required in order to later decide if the association between the two studied words is strong enough and if one without the other is considered as a semantically autonomous piece of information or not.

To achieve this, we catch the association between the input words in two steps: *merging adjacent words* and *enlarging keywords to co-occurrent ones*. The two steps are performed to satisfy the cohesiveness property of the final keywords. In our approach, we consider two words to be in the same window, if there are at most two non-stop words between them. We empirically found this length of window effective enough for our problem. However, this may not be necessarily the most optimal size of the window.

Merging adjacent words. The first step in generating keywords is to detect the top words which significantly appear as adjacent words within the studied web page and to

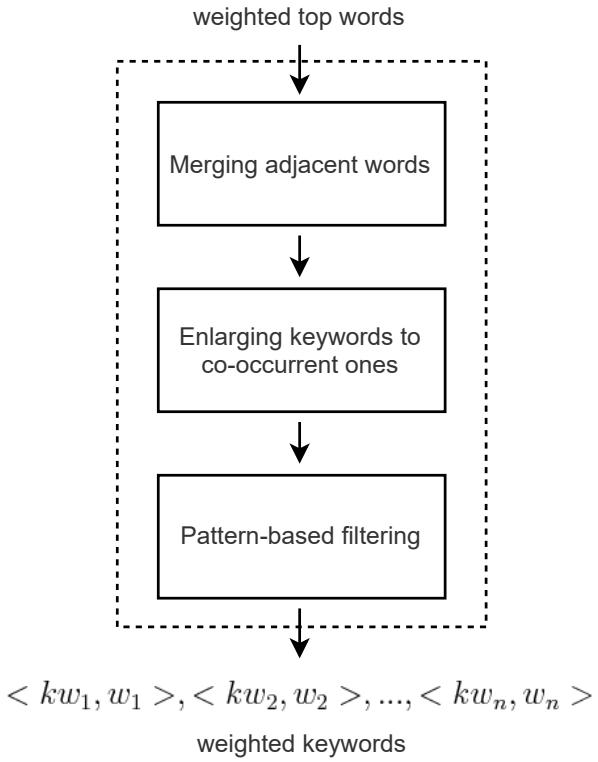


Figure 5.7: Different steps of the keyword generation

concatenate them. Since we aim at generating cohesive keywords and not any combination of words, we first merge adjacent words, which may not be semantically autonomous individually. These words may also have a different meaning when associated with another word than used independently. In this case, using their individual form in different combinations may lead to generating lexical units which do not correctly represent the subject of the studied document. Recall the example of "traffic issues in New York", where the meaning of the compound cannot be derived compositionally from the meaning of "new" and "york" taken independently. In this example, generating a combination like "traffic in York" brings wrong information about the discussed subject. Another example of strongly associated words is the numbers and their corresponding metrics. By "metrics", we mean units of measurements such as *m, cm, mm, kg*, etc. In the *<number, metric>* structure, combining either the number or the metric with other words of the content generates wrong combinations. To avoid this, we generate a single lexical unit for the *<number, metric>* structure and remove its individual words from the set of the top words to avoid merging them with other words.

It is also possible to have words which appear frequently both in adjacent and individual forms. In this case, sequences of words are generated but the initial words are also kept for generating more combinations in the next step. In general, the degree of dependency of two words is determined based on how often the words occur independently of each other. If this frequency is higher than a threshold, we infer that the word is semantically

autonomous *per se* and must be kept for further analysis. Otherwise, it cannot be used individually for generating keywords. Figure 5.8 shows an example of word sequences along with the decision on the dependency of their constituent words.

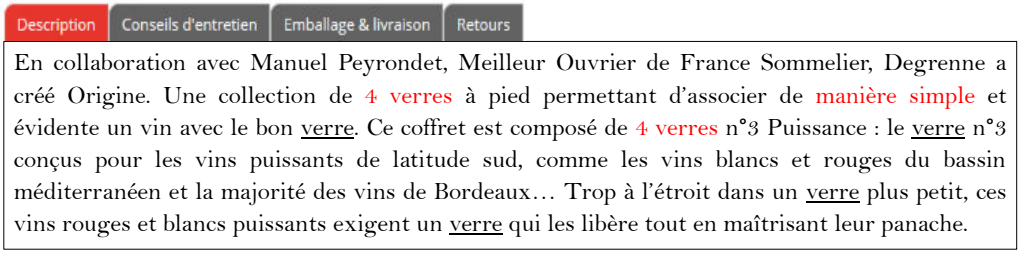


Figure 5.8: Example of word sequences

In this example, two word sequences are studied: "4 verres" and "manière simple". In the first case, the number "4" does not appear elsewhere in the content, whereas "verres" frequently appears as an individual word. As a result, this number is removed from the list of top words and its adjacent form is added to the list for further processing. In parallel, "verres" is detected as a semantically autonomous word and therefore is kept in the list. In the second case, both words "manière" and "simple" appear only within sequences of words. So, they are both removed from the list of the top words and their concatenation is added to the list.

When two words are merged, a weight is assigned to the word sequence, which specifies its importance within the studied document. This weight is computed based on the weight of the constituent words. There are different ways to aggregate the weights. In our approach, we use a simple average as aggregation function.

Enlarging keywords to co-occurrent ones. After generating the sequences of words and determining the dependency of their constituent words, we eventually expand them to longer keywords using the previously mentioned window-wise co-occurrence feature. Sliding the window over the content of the studied web page, the units which frequently co-occur in the same window form new combinations. Similar to the previous step, the degree of dependency of the constituent units is determined based on their frequency out of the studied combination. The weight of the new combination is computed by taking an average over the weights of the constituent units. We also make use of some heuristics. To avoid redundancy, we do not merge units with the same stem[5] even if they are significantly co-occurrent. As an example, we do not generate "experiment experience" as a combination, since "experiment" and "experience" both have the same stem "experi". To merge the co-occurrent units, we use prepositions so that to generate well-formed and cohesive keywords. Figure 5.9 shows an example of the combinations generated in this step. These combinations are considered as the candidate keywords of the studied web page. In this example, the following combinations are generated from the co-occurrent

[5]We use the Porter stemming algorithm for this purpose.

units: “collection de 4 verres”, “collection de 4 verres à pied”, “4 verres à pied”, “4 verres à pied de manière simple”, “coffret de 4 verres”. Since “4 verres” appears individually in different combinations, we consider it as a semantically autonomous unit and return it also as a candidate keyword. On the contrary, “collection”, “pied”, “manière simple” and “coffret” always appear as co-occurrent units and are not proposed as candidate keywords.

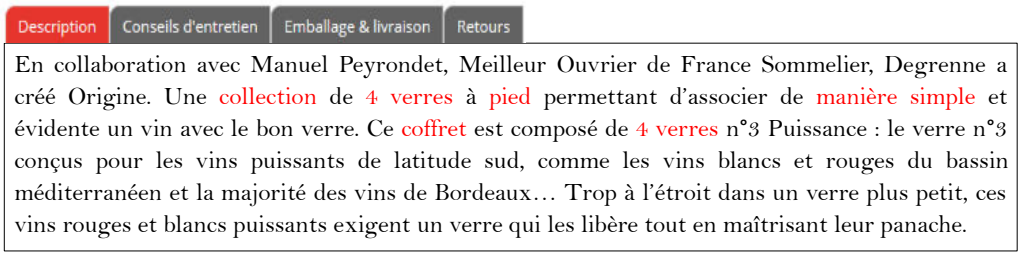


Figure 5.9: Example of the extracted units: “collection de 4 verres”, “collection de 4 verres à pied”, “4 verres à pied”, “4 verres à pied de manière simple”, “coffret de 4 verres”.

Pattern-based filtering. After generating the candidate keywords, a pattern-based filtering is performed on them in order to eliminate keywords which are not interesting for our application. These patterns have been obtained empirically by analyzing a wide range of keywords extracted from different websites. Some of these patterns use morpho-syntactic features, while some others simply correspond to uninformative sequences that we found frequent in our results. Following is the list of the patterns that we exclude from the final extracted keywords. Of course, these patterns could be different for a different application.

- Single words with parts of speech other than nouns. In our approach, we make use of different POS tags in order to generate cohesive keywords. However, at the end of the keyword generation step, adjectives, negative adverbs, numbers, etc., which appear as single words are not interesting to be kept as the final keywords of the studied page.
- Sequences strictly corresponding to the <*number, metric*> pattern, such as “24 cm”, without any additional word. We found such keywords too generic for our application and so we remove them. However, if the sequence includes other word(s), we consider it as an interesting keyword, *e.g.* “poêle 24 cm”.
- Keywords with information about dates and addresses. In our application, we are not interested in such keywords. Hence, we remove them from the final set of keywords, *e.g.* “exposition 2017”, “1 rue de Paris”, etc.
- Keywords that do not meet the keyword length requirement. We recall that in our approach, the length of keywords can be from 1 to 5 words, without considering the stop words. “collection de 4 verres à pied de manière simple” is an example of a long keyword that is removed in the filtering step.

At the end of this filtering process, the final set of extracted keywords is returned. These keywords are in lemmatized form. Since in our enrichment application keywords are recommended to users, we need to convert them into their original form so that only well-formed keywords are proposed to users. To do this, we compare the lemmatized keywords with the original content of the studied page. For each keyword, its most common form in the content is considered as the original form. These original keywords are the final output of the keyword extraction step. Figure 5.10 shows an example of a text along with its lemmatized and original extracted keywords.

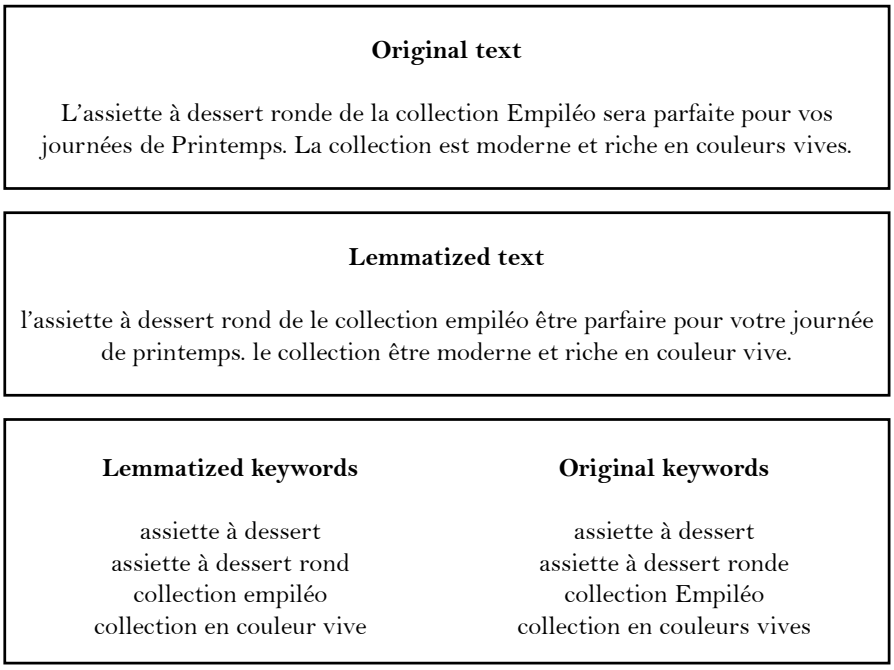


Figure 5.10: Example of the original and the lemmatized keywords

Depending on the target application, all the keywords extracted from a collection of web pages might be needed. In this case, keywords extracted from each page of the collection should be gathered and this could introduce redundancy to the list of keywords. The redundancy can be related to the duplicate keywords, that can be simply removed from the collection, but also to the ones which are morphological variants of each other. We refer to these keywords as *near duplicates* and consider two keywords to be near duplicate if they have one of the following properties:

- The same lemmatized form, *e.g.* "assurance voiture", "assurances voitures"
- The same lemmatized tokens but in a different order, *e.g.* "assurances voitures", "voiture assurances"
- The same normalized form after removing diacritic signs (non-English characters), *e.g.* "buche", "bûche"

- Uninformative uncommon tokens which are stop words in our approach, *e.g.* “piscines et accessoires”, “accessoires piscine".

Depending on the frequency and the score of near duplicate keywords, the less important one is removed from the collection.

Chapter 6

Keyword Extraction Evaluation

Contents

In this chapter, we present our experiments on the keyword extraction approach. We evaluate and discuss the results. Our keyword extraction approach consists of different steps, each of which can be evaluated individually but such an evaluation would be very demanding. Here, we focus on two points: comparing the effectiveness of the different co-occurrence scores explained in Section 5.2 and evaluating the quality of the extracted keywords. Other steps of the algorithm have been studied through side experiments, which guided our work but are not presented in this thesis. Among these steps, the extraction features are highly important. Although the choice of these features has been justified in our work, the contribution of each feature in extracting the main information of a web page remains to be done (the same analysis as in (Yih et al., 2006)).

We perform user-based evaluation: we ask a French native speaker, who is an expert in the target application and familiar with the target domains, to evaluate our results. Due to the numerous extracted keywords, we could not ask more than one evaluator to evaluate the results. On the other hand, we assume that one "expert" evaluator can be more reliable than several "non-expert" evaluators. As explained in Chapter 2, the user-based evaluation can be performed *a priori* or *a posteriori*. In our evaluation, we perform the latter one, where we firstly extract the keywords and then evaluate them without asking the evaluator to assign keywords to the studied pages *a priori*. In fact, we assume that assigning keywords to web pages is both complex and quite subjective. Since in designing the keyword extraction approach, *representativity*, *specificity* and *cohesiveness* were taken into consideration as the desired keyness properties, in the evaluation step, the evaluator is asked to assess the quality of the extracted keywords in terms of the same properties. In the first two cases, the goal is to verify if the keywords are good descriptors of the studied web page. A keyword is considered as a good descriptor if it discusses the main subject(s) of the page and if it is not too generic to appear frequently in web pages of various domains. As an example "next page" is not a good descriptor of a web page. As the third property,

the structure of the extracted multi-token keywords is studied to evaluate if they are well-formed. For example, "beauty products of" and "beauty buying products" are ill-formed variants of "buying beauty products".

To study the effectiveness of our approach with respect to the state of the art, we compare it with a baseline approach, the TF.IDF which is widely used for this purpose. However, our approach extracts both single and multi-token keywords, while the traditional approach of TF.IDF returns only single words. We therefore had to adapt the TF.IDF method to make its results compatible with ours. The exploited baseline approach is explained in more details in Section 6.3. We perform a similar user-based evaluation on the results obtained using TF.IDF. Eventually, the values of the evaluation measures obtained for the two approaches are compared.

Although our side experiments show that using an external resource, *i.e.* query log, slightly improves the performance of our approach, we do not exploit it in the experiments presented in this chapter, since we are mainly interested in evaluating the approach with knowledge-poor property.

Our keyword extraction approach was initially designed for French but in such a way that it could be easily tunable to other languages by setting a language-specific tagger and adapting the list of stop words and the units of measurements, such as *meter, centimeter,* etc. We implemented the approach in twelve languages: French, English, Spanish, Portuguese, Dutch, German, Estonian, Bulgarian, Russian, Polish, Italian, and Romanian. The first four languages have been already tested by native speakers on different web pages but we did the formal evaluation of our approach only on the French language, which is the native language of our evaluator.

In the following, we explain the details of the experimental data and the results.

6.1 Evaluating the co-occurrence scores in the top words selection

In this section, we perform different experiments in order to study the impact of the different co-occurrence scores presented in Section 5.2 on our keyword extraction approach. We recall that three of these scores are taken from the state-of-the-art, while the Weight-based Score (WBS), which takes the weights of the studied words into consideration, is ours. We also perform an experiment on the effective number of words to be selected after applying the co-occurrence score (the *top-K'* words, see Section 5.2).

We collect 200 random URLs from 10 different websites. For each URL, we execute our approach in two ways:

1. Exploiting only the extraction features to select the top words (*top-K* words), which are passed to the keyword generation step.

2. Merging the *top-K* words with the co-occurrent ones, *i.e.* the *top-K′* words, and passing them all to the keyword generation step.

The goal is to validate the second strategy using the schema presented in Figure 6.1. A score is considered to be effective if it generates as more as possible good keywords and as few as possible bad keywords. The keywords are evaluated in terms of representativity, specificity, and cohesiveness.

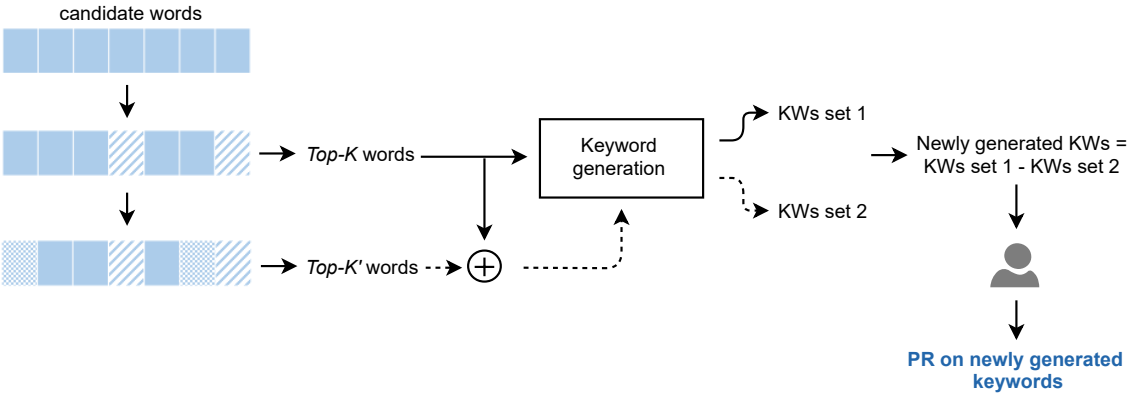


Figure 6.1: Schema for evaluating the newly generated keywords using the co-occurrence scores (KW=keyword, PR=precision)

In order to determine the effective number of words to select with the co-occurrence score (K'), we try two threshold values. The first one is inspired by Matsuo and Ishizuka (2004): we select 30% of the total number of the studied words. Clearly, this threshold is a function of the length of the studied page. The second threshold, however, is set to a fixed value of 10. In this experiment, our goal is to find an effective value for K', without claiming that it is the best one. As a future work, we could perform experiments on other values in order to adjust that threshold.

Table 6.1 shows the results of our evaluation on the different co-occurrence scores and using the two mentioned threshold values. The exploited evaluation measure is *precision* on the newly generated keywords (see Figure 6.1), which is obtained by dividing the number of new good keywords by the total number of newly generated keywords. According to the table, Weight-based Score (WBS) with the threshold value of 10 returns the highest precision. This confirms that word weight is an important factor to take into account in computing the co-occurrence score.

Table 6.1: Precision on the newly generated keywords

Co-occurrence score / Threshold	X^2-*measure*	*Improved* X^2-*measure*	$Score_{Rose}$	WBS
30% of the words	53%	55%	65%	65%
10 words	57%	60%	75%	85%

6.2 Evaluating the extracted keywords

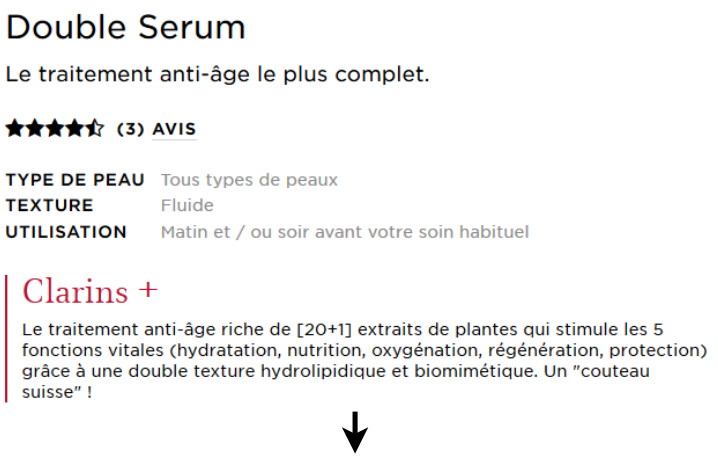
In this section, we study the quality of the extracted keywords. For this purpose, as before, we evaluate them in terms of three properties: representativity, specificity, and cohesiveness. A keyword is labeled as a "good" keyword if it satisfies all these properties with respect to the content of the studied page. In this evaluation, the ranking of the keywords is not studied *per se*, as it is very difficult for the evaluator to give ranked lists of keywords.

In our experiment, we take ten various websites into account. Usually, websites contain some pages that are very generic and not interesting for our enrichment application, such as "contact", "terms of use", and "shipping information" pages. To have a more precise evaluation and to target only interesting pages, we initially remove the generic ones. We extract keywords of each remaining URL using our approach. All the selected URLs along with their extracted keywords are passed to the evaluator, who labels the keywords as "good" or "bad" for the page. For the bad keywords, the evaluator has to specify the reason, indicating whether they are not good descriptors of the pages or they are ill-formed. This feedback from the evaluator could help us to improve our approach in the future. The underlying idea is that ill-formed patterns might be detected in the results and then used in a post filtering step to remove keywords that match these patterns or to change them into well-formed keywords. Figure 6.2 shows an example page with some of its extracted keywords and the evaluation labels assigned by an evaluator. We note that keywords have been extracted according to the associations of words in different parts of the page but the figure only represents the body of the page and not the other parts, such as title and Meta description.

To compare our keyword extraction approach with the state-of-the-art, we use TF.IDF as a baseline approach. TF.IDF is a traditional statistical approach, which has been widely used as a baseline approach in different works. However, this approach is mainly used for extracting single token keywords, while our approach aims at extracting both single and multi-token keywords. In order to have the same level of granularity in both approaches, we use TF.IDF as a feature for selecting words in the top words selection step. The extracted top words by TF.IDF are then passed to our keyword generation step and the final keywords are evaluated by the evaluator.

We generate our own experimental data for the evaluation task. Although it is always interesting to confront user-based evaluation obtained on real applications to benchmarks, we focus on the first evaluation approach, since due to the following reasons, the state of the art benchmarks are difficult to exploit for our problem:

- Robustness of our keyword extraction approach over various domains needs to be evaluated. To do this, an evaluation benchmark must consist of various domains. This is, however, not the case in many of the existing benchmarks.



Keyword	Evaluation label
double texture hydrolipidique et biomimétique	GOOD
traitement anti-âge complet	GOOD
plantes functions	BAD
traitement anti-âge	GOOD
double sérum	GOOD

Figure 6.2: Example page along with the extracted keywords and the evaluation labels

- The existing public benchmarks mainly contain scientific papers and cannot be used for evaluating our approach of extraction, which has been specifically proposed for extracting keywords from web pages.
- Most of the existing benchmarks are in English, while we specifically evaluate French keywords in our work.

In the following, we present the experimental data. We then present the evaluation result, including the comparison with the baseline approach.

6.2.1 Experimental data

In total, we have already executed the keyword extraction approach on roughly 2,000 French websites, approximately from 400 various domains, but we target only ten websites for the purpose of evaluation. To study the robustness of the approach in various domains, we pick the websites from very different domains. Table 6.2 summarizes the target websites and the number of the URLs in each website after filtering out the uninteresting pages. Due to confidentiality issues, we refer to the websites not by their names but by their domains. As the table shows, we study both small and big websites in our evaluation.

We execute our keyword extraction approach on each website. The number of the extracted keywords is a function of the representativeness and the importance of the words

Table 6.2: Target websites in the keyword extraction evaluation

Website	*Total number of pages*
Advertising business	70
Construction material	200
Financial Services	303
Beauty products	2,446
Insurance	1,906
Décoration	2,633
Certification	253
Industrial gaz	2,371
Electric bed	457
Recipes	856

within the studied pages and the way they are associated with each other. Web pages may have some extracted keywords in common. In addition, some keywords are morphological variants of each other, *i.e.* near duplicate keywords. Table 6.3 shows the number of the extracted keywords from each website after removing duplicate and near duplicate keywords. It also shows the average number of the extracted keywords per URL. Some of the studied websites in our experiment, such as "Beauty products" and "Décoration", contain a considerable amount of information and a lot of keywords are extracted from their corresponding pages. Near duplicate keywords in each website are detected using the heuristic explained in Section 5.3.

Table 6.3: Statistics on the keywords extracted using the proposed approach. The average number is calculated over all the keywords and not the unique ones.

Website	*#unique KWs*	*Average number of KWs/URL*
Advertising business	177	3.2
Construction material	503	5.9
Financial Services	758	3.6
Beauty products	13,465	9.6
Insurance	5,099	4.3
Décoration	14,728	9.1
Certification	761	4.1
Industrial gaz	11,097	5.9
Electric bed	1,994	6.2
Recipes	4,227	7.1

As will be seen in the following of this chapter, user-based evaluation is performed on the extracted keywords and precision value is computed accordingly. However, to compute the recall value, a gold standard set is needed. In the following, we explain the difficulties of this task and present our heuristic for generating this set.

Gold standard set for computing recall value

Unlike precision, computing the recall value is challenging due to the difficulties of generating a gold standard set of keywords for web pages. It is very demanding for the evaluator

to extract all keywords of each studied page and to generate the gold standard set accordingly. It is also not possible to exploit existing data for this purpose. For some pages, keywords in the *Meta keywords* tag of the HTML source codes could be used but many pages do not have this information or their Meta tags contain keywords which are not good descriptors of the pages.

Due to these difficulties in generating a gold standard set of keywords for each URL, we generate one for keywords of the whole website. To achieve this, we make use of four pieces of information: 1) the keywords extracted by our approach that have been evaluated as good keywords, 2) the keywords extracted by the baseline approach, labeled as good keywords by the evaluator (see Section 6.3), 3) the keywords introduced by an expert of the domains for the studied websites, 4) the keywords obtained from Google analytics tool that are well-formed and relevant to the website. In the last two cases, the keywords could have words which are out of vocabulary of the studied website. Due to this property, we may never reach the recall value of 100% on our extracted keywords. To better show the effectiveness of our approach in terms of the recall value, we remove the keywords which contain out of vocabulary words before computing this measure.

6.2.2 Experimental results and evaluation

We pass the generated experimental data to the evaluator, who studies the content of each URL and accordingly labels its extracted keywords as "good" or "bad". We recall that our evaluator is a French native speaker, who is an expert in the target application and familiar with the target domains. In this evaluation, the evaluator is asked to take all representativity, specificity, and cohesiveness properties of the keywords into consideration. For bad keywords, the evaluator is asked to specify the reason, mentioning if they are ill-formed or not good descriptors of the studied pages. The first reason is related to the cohesiveness property, while the latter one is linked to the representativity and specificity. In case of missing more than one property, the evaluator mentions the one which is more dominant. This information could help us to improve the approach in the future.

The evaluation measures in this step are *precision*, *recall*, and *F1-measure*. We compute the precision value on both web page and website level. In the former case, we first compute the precision on each web page of the website individually and we take an average over all the obtained precision values to have the average precision on the web page level. In the latter one, we aggregate all the extracted keywords and using their evaluation labels, we compute the precision on the whole website. It should be noted that duplicate and near duplicate keywords are removed in the aggregation step. Figure 6.3 illustrates the details of these two measures.

Web page level and website level precision values are not necessarily the same. The difference is that in the website level, all the duplicate and near duplicate keywords are removed. The website level precision could be higher or lower, respectively, if the removed keywords are mainly bad or good. Clearly, more duplicate and near duplicate keywords are

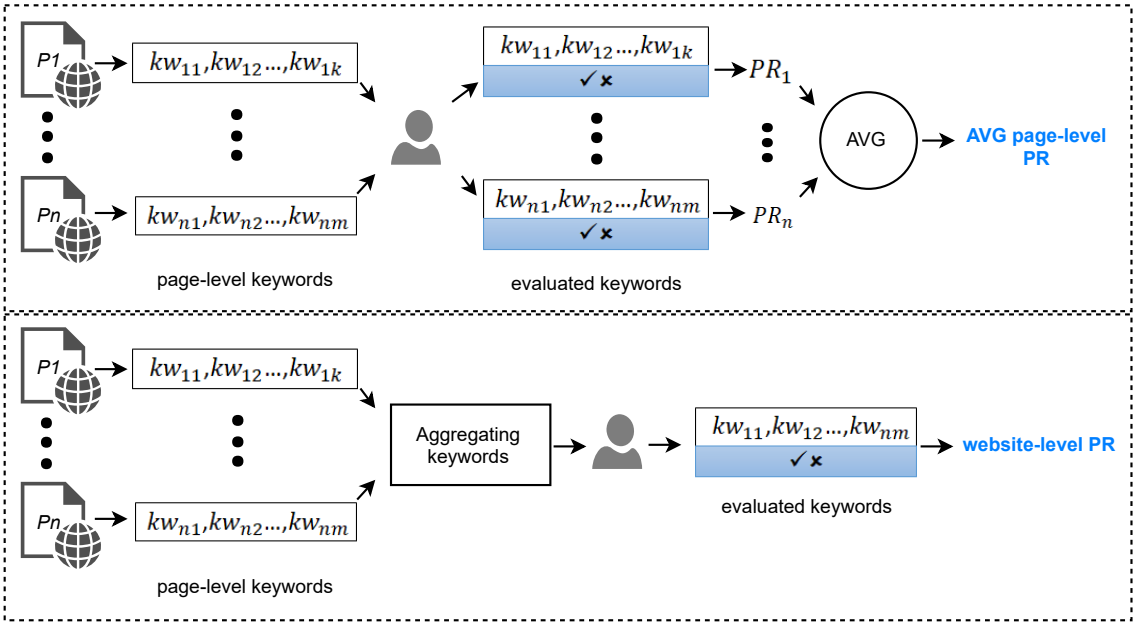


Figure 6.3: Schemas for measuring page-level and website-level precision values

extracted from websites in which many web pages have duplicate contents. In our experiment, "Construction material", "Beauty products", and "Décoration" respectively have the largest number of such keywords. On the other side, "Certification", "Advertising business", and "Industrial gaz" have more unique web pages and so more unique keywords. Due to the great number of the URLs targeted in this evaluation, we do not report the precision obtained for each web page. Table 6.4 reports the average web page level precision in each website.

Table 6.4: Web page level precision on the keywords extracted by the proposed approach

Website	*Precision*
Advertising business	81.31
Construction material	90.52
Financial Services	86.9
Beauty products	84.23
Insurance	90.46
Décoration	91.44
Certification	84.01
Industrial gaz	87.38
Electric bed	85.08
Recipes	88.67
AVERAGE	87

Due to the difficulties in generating a gold standard set of keywords for each URL, computing a "real" recall value on the web page level is not feasible. However, we compute the recall value on the keywords extracted from the whole website by making use of the gold standard set that we generated using different tools and approaches (see 6.2.1). We

applied a heuristic for generating this set without claiming that it contains all the keywords of the pages.

Using this gold standard set, we report the recall value over the extracted keywords of the whole website. The match between near duplicate keywords should be taken into consideration while computing the recall value. In other words, keywords which are morphological variants of each other must be matched when comparing the gold standard set and the extracted keywords. This comparison is performed automatically in our evaluation, but this match may not be fully captured by this automatic approach.

We note that due to the issues in generating a "real" gold standard set and in capturing the match between near duplicates, we rather compute an "approximate" recall value and consequently an "approximate" F1-measure.

Table 6.5 shows the values of the evaluation measures on the website level. Among the studied websites, "Recipes" has the lowest recall value. This is mainly due to the fact that the studied pages in this website are mostly long and very descriptive. We observed that the constituent words of these pages are mainly unique and they do not occur frequently in the content. In addition, their title and Meta description tags mainly contain very generic phrases. As a result, not all the discriminative words can be detected using our extraction features, which decreases the recall value. For instance, we miss ingredient words. As a future work, we need to study web pages in which words do not appear frequently in order to propose alternative extraction features.

In our enrichment application, recommending bad keywords to user is not acceptable. Therefore, precision is more important than recall. Table 6.5 also represents the values of $F_{0.5}$, which emphasizes more on the precision value.

Table 6.5: Evaluation results on the keywords extracted by the proposed approach on the website level

Website	*Precision*	*Recall*	*F1*	$F_{0.5}$
Advertising business	82.48	60.58	69.85	76.92
Construction material	94.23	61.24	74.23	85.06
Financial Services	84.69	64.07	72.95	79.57
Beauty products	87.59	76.54	81.69	85.13
Insurance	88.11	73.4	80.08	84.71
Décoration	93.48	77.4	84.68	89.75
Certification	83.83	62.61	71.68	78.51
Industrial gaz	90.16	72.27	80.23	85.91
Electric bed	81.99	77.74	79.81	81.1
Recipes	89.7	55.43	68.52	79.83
AVERAGE	87.62	68.12	76.37	82.64

Figure 6.4 shows the ratio of the keyness properties that have not been satisfied in the bad keywords extracted in all the ten websites. Specificity is related to the cases where the evaluator found the extracted keywords to be very generic, while in Representativity, the keywords were not related to the subject of the studied page. In Cohesiveness, the structure

of the multi-token keywords is reported as an ill-formed structure. We observe that bad keywords are mainly related to the specificity property. On the dataset of construction material, "barbelé fil", "choix de couleur" and "demande" are respectively examples of ill-formed, non-representative and too generic keywords extracted by our approach.

To reduce the ratio of "bad" keywords, as a future work, we should mainly study two points, which respectively target "cohesiveness and specificity" and "specificity and representativeness" properties:

- Improving patterns in the pattern-based filtering of keyword generation (Section 5.3): patterns of the ill-formed keywords need to be detected and considered in the keyword generation step so that cohesiveness property of the extracted multi-token keywords would be better satisfied. In addition, our analysis shows that some generic keywords have common patterns, *e.g.* "page 4" and "page 11". These generic patterns could be detected and added to the pattern-based filtering in order to extract more specific keywords.
- Performing more analysis on the extraction features: some generic keywords have no common patterns for filtering. To avoid extracting such keywords, the extraction features should be improved in order to extract more discriminative words from the web pages and consequently to generate more specific keywords. In addition, having more effective features, the extracted keywords would be more relevant to the content of the studied pages and the representativity property of the keywords would be better satisfied.

6.3 Comparing with a baseline approach

Similar to our work, Yih et al. (2006) target web pages as the input data. The two works share some extraction features, although our work makes use of a smaller set of

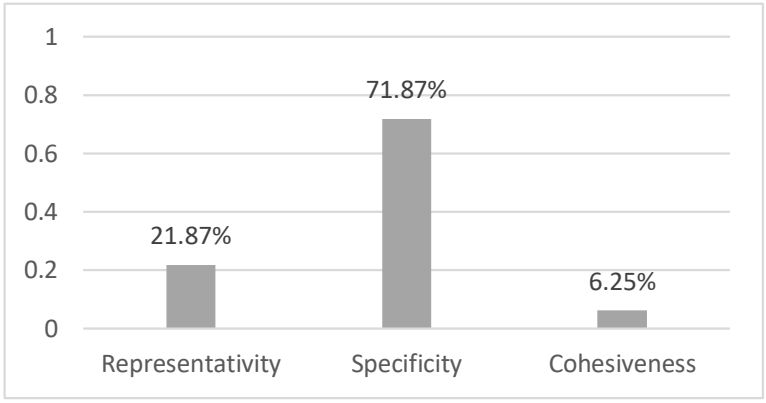


Figure 6.4: Analyzing the "bad" keywords extracted by the proposed approach

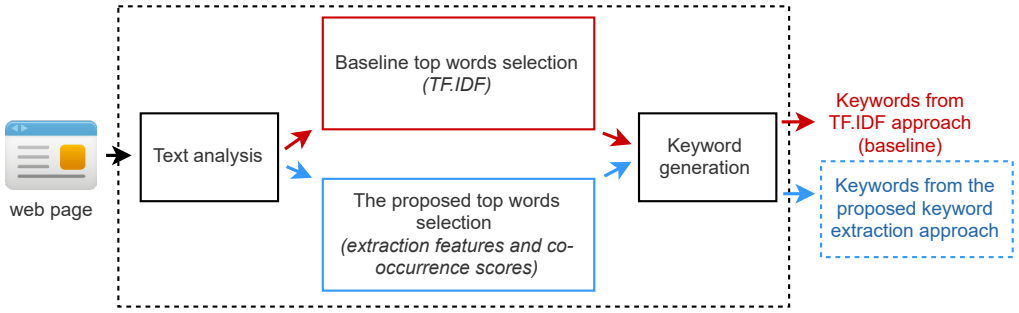


Figure 6.5: Different steps of the baseline (top) and the proposed (bottom) keyword extraction approaches

features. Due to this similarity, it would be interesting to compare our approach with theirs. However, neither their evaluation corpus nor their method is available and so such a comparison is not feasible. Alternatively, we choose TF.IDF as the baseline approach, which is a traditional and widely-used statistical approach for this purpose.

Applying TF.IDF on the page content leads to the extraction of single token keywords. In our approach, however, we extract both single and multi-token keywords. In order to have the same level of granularity in our evaluation, we exploit TF.IDF in a way to extract both single and multi-token keywords. We recall that our keyword extraction approach consists of three main steps: text analysis, top words selection, and keyword generation. As the baseline approach, we use TF.IDF only for the top words selection step. In other words, we have the same procedure of text analysis and keyword generation in our approach and the baseline approach. Figure 6.5 illustrates the steps in the two approaches. It should be noted that to compute the IDF values while extracting top words of a web page, we make use of all pages of the corresponding website as the corpus of documents.

To compare with the baseline approach, the same data as in Table 6.2 is exploited. The number of the extracted single words in our approach is a function of the representativity and the importance of the words within the studied page. In the baseline approach, the same number of words as in our approach is selected and passed to the keyword generation step. Table 6.6 shows the statistics on the keywords extracted using the baseline approach.

Although the number of the single words passed to the keyword generation step is equal in the two approaches, the number of the generated keywords may not be the same, as it depends on the way that the selected words are associated in the content of web pages. However, since TF.IDF assigns higher weights to words which are less frequently used in the corpus and focuses more on the specificity feature of words, it generates a smaller number of duplicate and near duplicate keywords.

Similar to the evaluation of our approach, here, we compute the precision on both web page and website levels but the recall value only on website level. Table 6.7 compares the web page level precision values obtained for the baseline and the proposed approach.

Table 6.6: Statistics on the keywords extracted using the baseline approach. The average number is calculated over all the keywords and not the unique ones.

Website	*#unique KWs*	*Average number of KWs/URL*
Advertising business	172	2.9
Construction material	1134	5.05
Financial Services	651	2.9
Beauty products	14060	8.09
Insurance	5471	4.3
Décoration	14411	8.04
Certification	842	4.05
Industrial gaz	9162	4.8
Electric bed	1952	5.8
Recipes	4137	6.8

Table 6.7: Web page level precision on the keywords extracted by the baseline and the proposed keyword extraction approaches

Website	*Precision - Baseline (TF.IDF)*	*Precision - The proposed approach*
Advertising business	46.82	81.31
Construction material	54.6	90.52
Financial Services	53.97	86.9
Beauty products	60.36	84.23
Insurance	56.01	90.46
Décoration	59.24	91.44
Certification	57.36	84.01
Industrial gaz	55.19	87.38
Electric bed	45.18	85.08
Recipes	57.07	88.67
AVERAGE	54.58	87

The website level precision, recall, and F1-measure are reported in Table 6.8. In both web page and website levels, our proposed approach performs considerably better than the baseline approach, specifically on the precision value. Figure 6.6 illustrates the comparison between the two approaches on the website level and over the ten websites.

We also analyzed the bad keywords extracted by the baseline approach in order to find the keyness properties that are not captured properly by this approach. Figure 6.7 shows the result of this analysis. On the dataset of construction material, "2000 mm de large", "première partie" and "commande" are examples of the keywords extracted by the baseline approach, which were respectively labeled as ill-formed (incomplete), non-representative and too generic keywords by the evaluator.

According to Figure 6.7, in the baseline approach, the bad keywords are mainly irrelevant to the content of the studied pages and so do not satisfy the representativity property. Comparing the percentage of the non-representative keywords extracted by our approach and the baseline approach (21.87% *vs.* 80.65%), we conclude that TF.IDF, on its own, is not an effective feature for capturing representative words of web pages. We believe that this is mainly due to the fact that TF.IDF considers the content of a web page as

Table 6.8: Evaluation results on the keywords extracted by the baseline approach on the website level

Website	*Precision*	*Recall*	*F1-measure*
Advertising business	44.18	31.53	36.8
Construction material	55.96	53.35	54.62
Financial Services	50.84	33.03	40.04
Beauty products	61.16	55.76	58.31
Insurance	56.42	50.43	53.26
Décoration	61.4	49.74	54.96
Certification	56.41	46.61	51.04
Industrial gaz	57.67	38.16	45.93
Electric bed	44.51	41.32	42.86
Recipes	55.32	33.46	41.7
AVERAGE	54.38	43.33	47.95

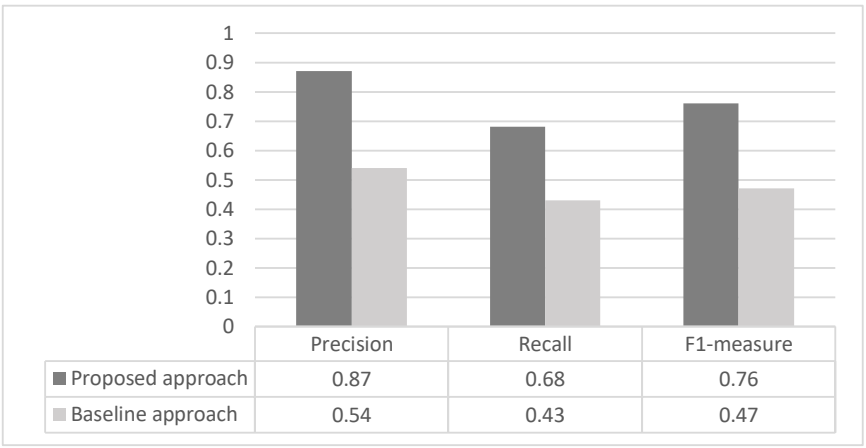


Figure 6.6: Effectiveness of the proposed approach *vs.* the baseline approach on the website level and over 10 websites

a plain text, where all parts of the page are equally taken into consideration. In other words, it does not exploit informational features which play an important role in detecting representative words of a page.

In our experiment, we apply the TF.IDF on document level and not on domain level. This means that the TF.IDF assigns low weights to words which appear frequently in different pages of a website. These words may, however, be good representative of the domain of study and assigning low weights to them could lead to missing representative keywords of the studied document. Since in our application, we are interested in representative keywords and not merely the specific ones, an alternative solution would be to apply TF.IDF on domain level in order to detect domain-specific keywords more effectively. We, however, did not try this solution because applying TF.IDF on domain level would require processing a large set of documents from different domains. In our recommendation ap-

proach, such a set may not be available. In addition, we did not find the complexity of this processing to be efficient for our application.

We, however, are not interested in generic keywords, which are common across different domains. Compared to our approach, we observed that TF.IDF can better avoid generating such generic keywords. As before, the cohesiveness property, corresponding to the keyword generation step of our proposed approach, is mostly satisfied.

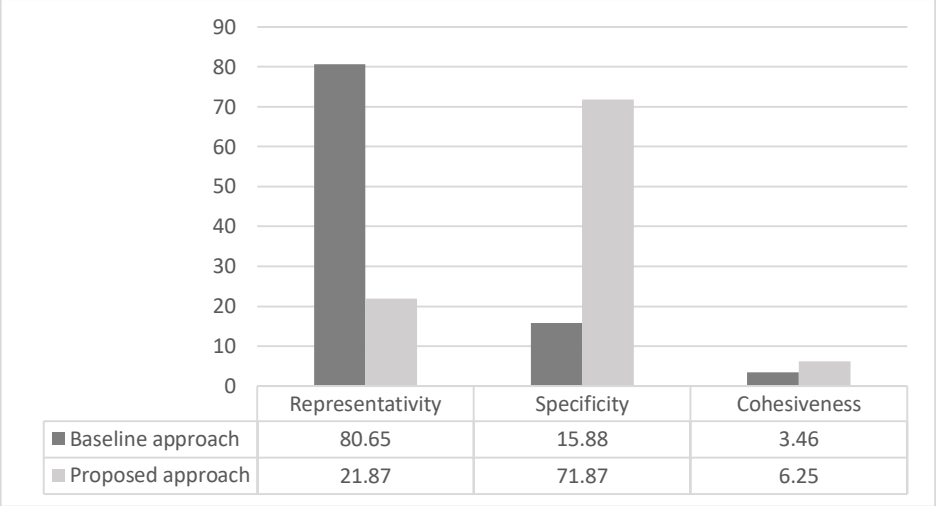


Figure 6.7: Properties of the "bad" keywords extracted by the baseline approach *vs.* the proposed approach

6.4 Conclusion

In this chapter, we performed an experiment on the number of single words selected in the second phase of top words selection step and we found the fix value of 10 to be effective enough in this phase. To sum up, in the first phase of top words selection step, depending on the representativity and importance of words within the studied document, *i.e.* their scores, various number of words are selected (*top-K*). In the second phase, this set is expanded by adding a fixed number of single words, *i.e.* 10, which frequently appear with the *top-K* words. Eventually, depending on the total number of single words and the way they are associated in the content of the studied document, various number of keywords are generated in the keyword generation step.

We also evaluated the effectiveness of the proposed keyword extraction approach. The obtained values for the evaluation measures indicate that our basic and knowledge-poor approach can effectively extract keywords from different web pages. More specifically, our approach achieved a high precision value (approximately 87% on both web page and web site levels), which is required for our enrichment application. Depending on the studied websites, values of the evaluation measures vary. However, the approach performs effectively on different domains and this confirms its domain-independent property.

Comparing to the baseline approach, the keyword extraction approach performs considerably better (54% *vs.* 87%). This indicates that our extraction features and the exploited co-occurrence score significantly outperform the TF.IDF feature. This could be due to the fact that TF.IDF is a statistical feature. This baseline approach analyzes a web page as a plain text without prioritizing some parts of the page using informational features. In addition, applying TF.IDF on document level leads to assigning low weights to common but representative words of the studied domain.

It is worth mentioning that the complexity of the two approaches is comparable and both are executed in a reasonable time.

Part V

Topic detection

Chapter 7

Topic Detection Methodology

Contents

Topic detection aims at detecting the latent topics in a collection of keywords. A collection of keywords may actually cover several topics. Some of these topics might be semantically related, while some others might be independent. In addition, a collection may contain ambiguous keywords with different meanings belonging to different topics that must be distinguished. Considering these issues, in this chapter, a topic detection approach is proposed in order to disclose the topics of a collection[1].

The topic detection approach takes a set of weighted keywords as input. It consists of two functions that return a two-level topic description so as to offer a more structured and a more informative recommendation. Topics in each level are represented as sets of keywords.

- *Coarse-grained topic detection*: This function aims at detecting the coarse-grained topics within a collection of keywords, detecting polysemies, and eventually returning the topics which are related to the enrichment's point of view. Those topics usually contain a large number of keywords, which are mainly expected to be "semantically related".

- *Fine-grained topic detection*: This function further divides the domain of study by dividing each relevant coarse-grained topic into sub-topics and generates a set of fine-grained topics. Keywords in those topics are mainly expected to be "semantically similar". Each fine-grained topic is identified by a representative keyword and a ranked list of keywords. The ranking shows the importance and the relevancy of the keywords within each topic. The representative keyword, which is the highest ranked keyword in the topic, shows its main subject. Having representative keywords makes the recommendation more informative and more understandable for users, who can

[1] Submitting a patent on the topic detection methodology is currently being discussed.

easily pick the desired subjects within the domain of study. It is also essential to have a ranked list of keywords in the recommendation, as our approach needs to make a balance between the reduction of the semantic gap and the length of the enriched document. The ranking allows users to add the recommended keywords to an input document in order of importance and relevancy to the corresponding point of view until the allowed maximum length of the document is achieved.

For each function, a graph-based approach is proposed in order to model explicit semantic relations between keywords. In this context, topic detection amounts to community detection with a graph of keywords. A graph-based approach consists of two main components: *graph generation* and *graph analysis*. The details of the two components is, however, different in the two functions. Figure 7.1 illustrates the overall framework of our topic detection approach. The coarse-grained and fine-grained functions are respectively shown as *Cg-TD* and *Fg-TD*. The fine-grained topic detection has two levels, both graph-based (not represented in the figure). In the first level, semantic similarity between keywords is analyzed to return sets of semantically consistent clusters, while in the second level, we analyze dissimilarity between keywords of different clusters to eventually generate discriminative clusters, *i.e.* fine-grained topics.

To sum up, three different graphs are generated in our topic detection approach: one for detecting coarse-grained topics and two for detecting fine-grained ones. We note that our topic detection approach is domain-independent. It was initially proposed for French but it is easily tunable to different languages. In the following, each function of the approach along with its corresponding components is explained in more details.

7.1 Coarse-grained topic detection

The first function in the topic detection step aims at detecting the underlying topics in a collection of keywords. We recall that in this thesis, a "topic" is defined as a set of keywords which can be semantically similar, such as synonyms, or semantically related, such as meronyms and hyponyms. The coarse-grained topic detection function also disambiguates the polysemous keywords. Since these topics are mainly generic and contain a large number of constituent keywords, we name them *coarse-grained topics*.

In order to detect the coarse-grained topics, we make use of a graph-based model. Unlike latent approaches, explained in Section 3.3, graphs are able to model explicit relations between keywords of a collection and they can take various types of relations into account. Graph analysis approaches let us obtain significant information about the keywords and their connections.

In the following, we explain the two components of the graph-based approach that we proposed for detecting the coarse-grained topics. The first component generates a graph of keywords and models their interaction according to their degree of similarity. In the second component, the generated graph is analyzed to identify the various topics that compose it.

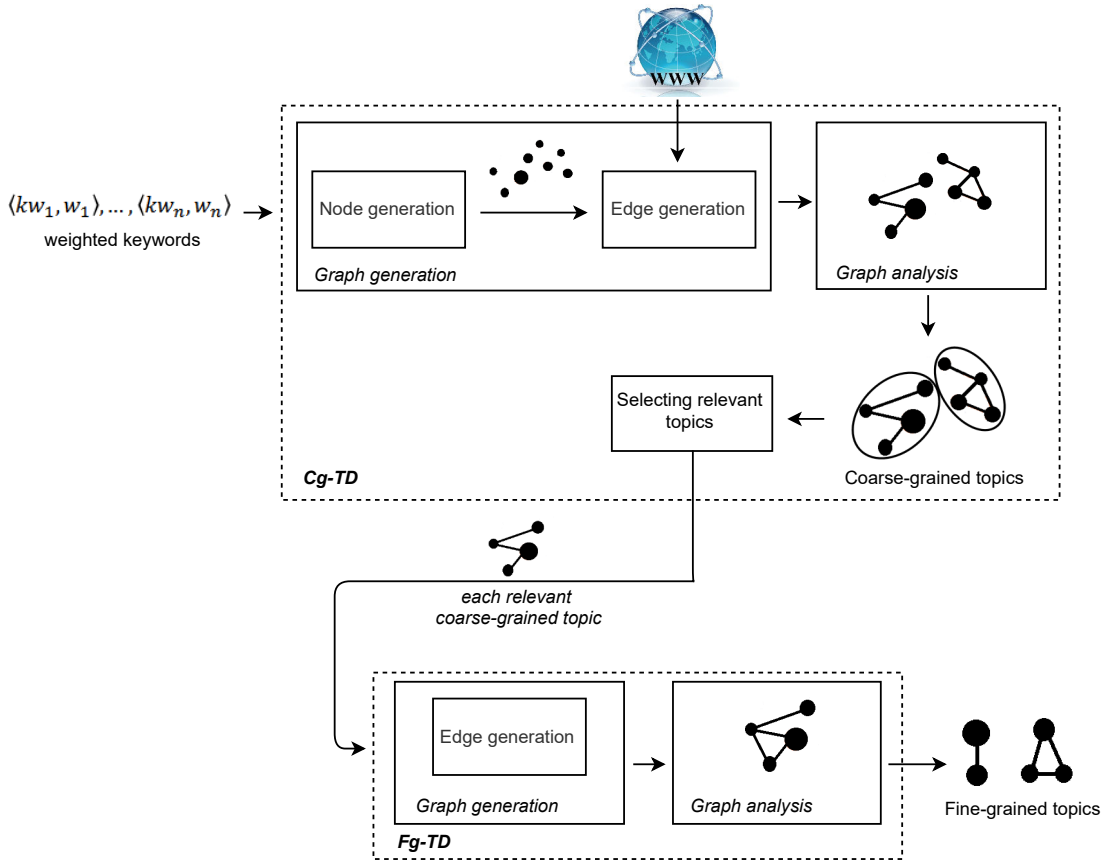


Figure 7.1: Topic detection framework

7.1.1 Graph generation

The first component in the coarse-grained topic detection generates a graph of keywords. By modeling the semantic relatedness between the keywords, we aim at detecting clusters of highly connected keywords in the generated graph, which are considered as our topics. In this work, we consider the graph as undirected and unweighted. Having an undirected graph, the similarity between nodes is assumed to be symmetric, which is a typical assumption behind most of the topic detection approaches. Although this is not always a true assumption, it simplifies the problem. By considering the relations to be symmetric, the computational cost of analyzing the graph decreases significantly. In addition, we found the evaluation of symmetric relations to be less complex and more precise than for asymmetric relations.

The degree of similarity between two keywords helps us to decide if they should be related in the graph. However, once this decision is made, we consider the graph as unweighted. The nodes of the graph are weighted but not the edges.

The graph generation starts by the node generation phase. It is then followed by the edge generation phase, where the semantic relatedness between nodes of the graph is modeled as edges of the graph.

Node generation

As previously mentioned, the first step of our graph-based approach is to generate a graph of keywords. In the node generation phase of the graph generation step, we take the weighted keywords, extracted from the collection of documents, as the input. These keywords may, however, be duplicates or morphologically variants of each other, *i.e.* near duplicates. They may also contain information which we did not find useful for our application. Brand names are examples of such uninformative information. In order to reduce the size of the graph, we apply two steps on the input keywords: *removing duplicate and near duplicate keywords* and *removing brand keywords*. As the output, we will have a set of unique keywords which are informative enough for our application. These keywords form the nodes of the graph. In the following, we explain each step in more details.

Removing duplicate and near duplicate keywords. Since keywords are extracted from different documents of the collection, the input list of keywords may contain duplicates. Since the graph nodes must be unique, the duplicate keywords should be removed. Each duplicate word has its own individual weight. While removing the duplicates, these weights need to be transferred into a representative weight that is assigned to the unique keyword which is kept and modeled as a graph node. To calculate this weight, we simply take an average over all the normalized weights of the duplicate keywords. Figure 7.2 shows an example on how the weights are re-calculated when redundant keywords are removed.

URL	Keyword	Weight
www.url1.com	KW1	2
www.url1.com	KW2	3
www.url2.com	KW1	1
www.url2.com	KW3	6

↓

URL	Keyword	Normalized weight
www.url1.com	KW1	0.666
www.url1.com	KW2	1
www.url2.com	KW1	0.166
www.url2.com	KW3	1

↓

Keyword	Final weight
KW1	0.416
KW2	1
KW3	1

Figure 7.2: Example of re-calculating weights while removing duplicate keywords

In addition to duplicate keywords, some keywords might be morphological variants of each other, called *near duplicates*. In this step, we remove these keywords to ensure that each node of the graph brings enough information, which is not found in any other node. In addition, by removing these keywords, we reduce the size of the graph and so the complexity of its analysis.

Our heuristic for detecting near duplicate keywords has been explained in Section 5.3.

Removing brand keywords. Depending on the target application and the type of the target documents, some of the extracted keywords may not be informative. In the node generation phase, the uninformative keywords should be removed. For our enrichment application based on web pages, we focus on brand names[2]. Although brands might be interesting for some applications, we did not find them informative for our specific problem as they are not useful for enrichment. In fact, they are too specific to be used for enriching the content of a web page with respect to other pages.

To overcome this issue, brands of a collection should be detected and keywords which contain these brands should be filtered out. Different approaches can be used to detect brand names. Depending on the domain of the target documents, one may use a relevant database of brands. An example is the database provided by *INPI*[3], the national intellectual property office of France. Although this database can be used as a source of brands in our approach, its big size and also the limitation on the number of requests to the database add complexity to our approach. Since the enrichment approach proposed in this thesis is knowledge-poor, here, we propose a knowledge-poor approach for brand detection, which uses documents of the enrichment collection to detect the brands. Although this approach may not detect all brands of the collection, due to its low complexity, we found it efficient enough for our problem. As a future work, we can work on a more effective approach of brand detection for the node generation phase.

For brand detection, we assume that the main brands of the collection mostly appear in the web page URLs of the collection. More specifically, they appear in the domain name of the URLs. Since this assumption is not always correct, our approach may not detect all the brands. However, as most of the brands can be detected making this assumption, to simplify the problem, we take advantage of it. To be clearer, the basic format of a URL, including its domain name, is shown in Figure 7.3.

Having the domain names of the URLs in the collection, brands are detected using the information provided by an online dictionary[4]. We assume that a domain name is a brand if it is not listed in the dictionary. We make this assumption because some domain names are meaningful and are not brands in other contexts. Hence, considering them as brands and removing their corresponding keywords leads to missing non-brand keywords. As an example, having "extension en bois" as the input keyword, one of the returned URLs

[2]By definition, brand is "a type of product manufactured by a particular company under a particular name"

[3]https://www.inpi.fr/fr

[4]Larousse online dictionary in our work

Figure 7.3: Basic format of a URL

by Google is `http://www.bois.com/renover/extensions`. It is obvious that the domain name of this URL, *i.e.* "bois", is not a brand and is one of the main words of the discussed topic. In order to determine if a domain name is a brand, it is sent as a query to the online dictionary and the result is parsed. If the dictionary detects the query and returns a definition for it, the domain name is detected as a non-brand. Otherwise, it is added to the list of the detected brands.

Figure 7.4: Example of the brand detection approach

It should be also noted that a domain name may have different variations and the brand detection approach should be able to detect these variations. A domain name may contain a dash punctuation mark ("-") in the URL of a web page but this mark can be changed into a space in the content of the page. An example is `https://www.gtf-bois.com/` with "gtf-bois" as the domain name. In the content of this page, the domain name appears both as "gtf bois" and "gtf-bois". Hence, keywords which contain "gtf bois" should be also considered as brand keywords. Due to this variation in the content of the URL and the page, we also check the meaning of a domain name's variations. In our approach, the variations are obtained by replacing the dash by a space and removing the dash (replacing it by no character). It should be noted that in case of having a multi-token domain name, it is likely that the dictionary returns no corresponding definition for a meaningful compound word. To overcome this problem, for multi-token domain names, the presence of each token in the dictionary is also checked separately. If all the tokens are present in the dictionary, the domain name is assumed to be non-brand. Example is `http://`

www.maison-ecologique-bois.com/, with "maison-ecologique-bois" as the domain name and "maison ecologique bois" as one of its variations. Although there is no entry in the dictionary for "maison ecologique bois", all the tokens can be found in the dictionary and the term is detected as a non-brand keyword. As before, "maison-ecologique-bois" is detected as a brand and any keyword which contains this brand is filtered out.

Figure 7.4 demonstrates the brand detection procedure for an input keyword "poêle à bois". The URLs of the collection are processed one by one and the first URL is shown as an example in the figure. For each URL, its domain name is firstly extracted and added to the list of "potential brands" along with its variations (if any). Having all the potential brands, they are passed to a dictionary and according to the result of the requests, they are detected as brands or non-brands.

At the end of the node generation phase, a reduced list of weighted keywords is returned, in which the keywords are presumably informative and different of each other.

Edge generation

Once the potential nodes are identified, one has to model their connectivity. As explained in Chapter 3, edges can represent different types of relations. In our approach, we aim at capturing the semantic relatedness between nodes in order to eventually detect sets of semantically consistent keywords, which are referred to as "topics" in this thesis. Since the keywords do not necessarily share common words, morphological similarity measures, such as Cosine and Jaccard, cannot effectively capture the semantic relatedness between them. In addition, in case of polysemies, the morphological measures detect similarity between keywords which are from different domains. Here, we propose an approach for capturing similarity without depending on the notion of common words between two keywords.

Since we aim at proposing a domain-independent approach of topic detection, knowledge-based similarity measures are not exploited in our work. Instead, we make use of corpus-based similarity measures. Unlike knowledge-bases, text corpora are updated more regularly and they contain more real-world relations. In our approach, two corpus-based similarity measures are proposed for capturing similarity between keywords. As a corpus, we exploit the web, which is a huge, multi-domain and multilingual resource that is updated regularly. Since new words are added to the web frequently, it is a good resource for mining semantic relationships for unseen words. The web also contains both common words, found in news articles, forums and blogs, and specific terms, found in scientific documents. However, the size of the web data is huge and it may contain unreliable or irrelevant content that should not be used by our approach. To mitigate this issue, we filter the web data associated to each keyword using search engines: each input keyword (node) is queried in a search engine and the returned result by the search engine is associated to the keyword as its *context*. We assume that due to the high performance of search engines, the generated context is highly relevant to the keyword and can be used as a source of information for capturing the similarity of the keyword with any other keyword.

Hence, our measures are knowledge-poor but they depend on search engines' results. Due to the limitations on the number of requests to search engines that can be made, we need to assure that the measures perform effectively with an optimal number of requests.

The context of each keyword is generated in the *context acquisition* step and it is then exploited for measuring similarities: *vocabulary-based similarity* and *co-occurrence-based similarity*. Each of these measures captures a specific aspect of similarity between keywords. The first measure is based on the context similarity between two keywords. It assumes that the contexts of similar keywords are close to each other in terms of their common words. The second measure, however, assumes that if two keywords are similar, it is more likely to have them as co-occurring keywords in different contexts. In Chapter 8, we evaluate the accuracy of each of these measures in distinguishing similar pairs of keywords from dissimilar ones. We then show that exploiting both of these measures outperforms using each of them individually.

We recall that we are interested in capturing the semantic similarity but also the semantic relatedness between keywords. We aim at capturing synonyms, meronyms and hyponyms. Unlike some works which assume antonyms to be semantically related, we are not interested in capturing this kind of relation for our enrichment problem. Our assumption is that the semantic similarity and the semantic relatedness are respectively captured through vocabulary-based and co-occurrence-based similarities.

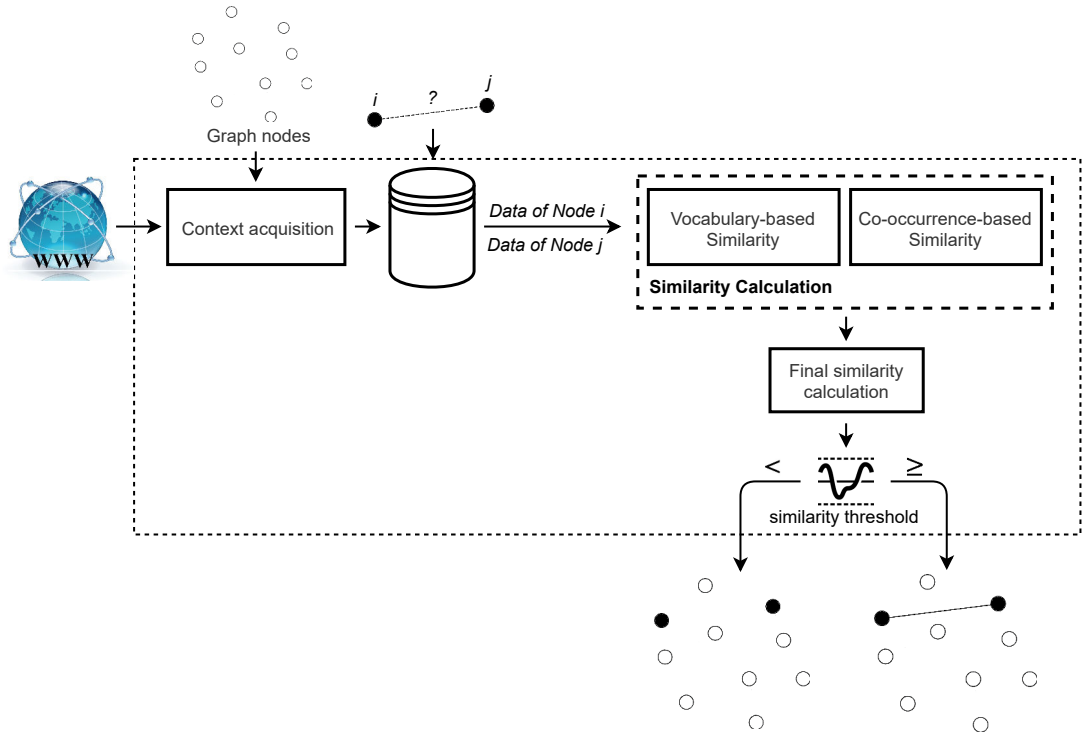


Figure 7.5: Edge generation flowchart

Context acquisition The similarity between any two nodes (keywords) is computed based on their corresponding contexts. For each keyword, a context is a collection of texts

which is returned by a search engine in response to querying that keyword. In more details, to compute the similarity between two given nodes, each node is queried individually in a search engine. The *top-P* web pages returned by the search engine are collected. The same constraints as in the enrichment collection generation step are applied in this step, *e.g.* removing pages issued from dictionaries, social networks, etc. Similarly, we only make use of organic results in the search engine result page (SERP), since they contain more informative and less biased content comparing to paid results. Having organic results of the search engine, snippets, page contents, and domain names of the URLs are extracted to be later used for calculating the similarity between any pair of nodes. "Snippet" is the short content which appears underneath the title of a page in the SERP. It is usually the content of the Meta description, a HTML tag that summarizes a page content. A search engine may however select a different text as the snippet in case that the Meta description is empty. Figure 7.6 shows an example of a snippet in the search engine result page.

We recall that our proposed approach is knowledge-poor but dependent on search engines. The complexity of the context acquisition step depends on the number of the graph nodes, *i.e.* queries, and the size of the SERP that is retrieved as context. In our work, we control this complexity by limiting the number of documents in the enrichment collection and also optimizing the number of requests sent to a search engine.

Any search engine can be used for generating the context. However, due to the wide use of Google, we exploit only this search engine for retrieving the required contexts.

Vocabulary-based similarity Vocabulary-based similarity is one of the similarities that we proposed for generating edges of the graph. The main assumption behind it is that similar keywords are expected to discuss the same subject and so are more likely to be used in closer contexts. Here, we consider the vocabulary of a keyword as its context and we consider that two keywords are very similar if they are associated to the same vocabulary or to overlapping ones. Often in this thesis, we exploit search engine results as a knowledge source. Here, we exploit the snippets returned by a search engine in answer to a keyword query as a context associated to that keyword. We can thus measure the similarity of two keywords through the similarity of their snippets and the overlap of their vocabularies. Snippets contain descriptive contents about the queried keyword. In addition, they are short and so cheap to analyze. Snippets are basically written in such a way that they contain the most relevant words to a query. Hence, comparing to page contents, the number of irrelevant words is considerably lower in snippets. Due to these

Figure 7.6: Example of a snippet in the search engine result page

advantages, we use them in order to expand the vocabulary associated to any keyword and for our vocabulary-based similarity. This expanded vocabulary gives a richer information for computing the similarity between any two keywords. The idea of expanding the vocabulary of the studied keywords is inspired by Sahami and Heilman (2006). However, our similarity is different from theirs both in the exploited source of information and the details of the similarity measure. In Section 8, we show that our measure outperforms the one proposed by Sahami and Heilman (2006).

The inspiration was taken from the experiments of Bollegala et al. (2007), who found their measures to perform better than the approach proposed by Sahami and Heilman (2006). Their measures rely on Google page counts. Some articles discuss that page counts returned by Google are not reliable anymore[5]. Our preliminary experiment on these measures confirmed this argument, as they did not capture the similarity between keywords effectively. In fact, functionalities of search engines, such as their page counts, are changing over time and measures which rely on these functionalities may work only for a certain period of time. According to the experiments presented in (Bollegala et al., 2007), after the page count-based measures, the approach of Sahami and Heilman (2006) performs the best in terms of precision and F1-measure. Relying on this finding, we were inspired by this work.

In more details, to compute the vocabulary-based similarity between any two nodes of the graph, we query each node in Google search engine. For the *top-P* results, the snippets are then extracted, merged and considered as a description of the queried node. Querying two nodes individually in the search engine, the unique words in their snippet sets are respectively called *extracted vocabulary 1* and *extracted vocabulary 2*. To be clearer, Figure 7.7 illustrates the vocabulary generation for an example query. We note that the difference between a snippet set and its corresponding vocabulary is that the former one may have duplicate words, while a vocabulary has only unique words.

After obtaining the vocabularies of the two keywords, a pre-processing is applied on the content of each vocabulary, including lemmatization, stop word removal, domain name removal, etc. The same approach of brand detection that was explained in the node generation phase is also performed on the two vocabularies in order to get rid of brand names, that we do not find informative for our application. The two pre-processed vocabularies are then compared based on their constituent words. We tried eight different measures to compute the similarity between two vocabularies according their common tokens. The measures are divided into two different categories: *unweighted measures* and *weighted measures*. Unlike the unweighted measures, the weighted measures assume that words of a vocabulary have different levels of importance. Hence, depending on the importance of the common tokens in each vocabulary, they have different contributions in these similarity measures. In Chapter 8, we compare the performance of the eight measures.

Following is the description of each of these measures. *Extracted vocabulary 1* and *extracted vocabulary 2* are respectively represented as $V1$ and $V2$.

[5]https://searchengineland.com/why-google-cant-count-results-properly-53559

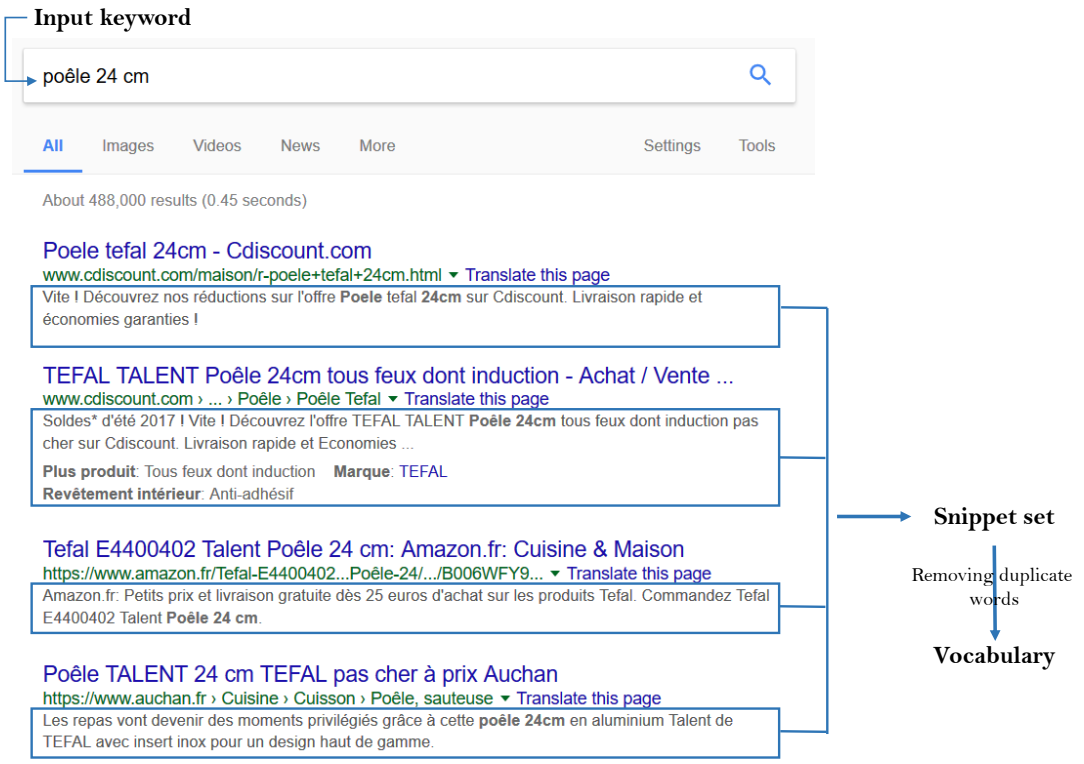


Figure 7.7: Example of the vocabulary generation

- *Unweighted similarity measures*: We tried six different similarity measures, which assume that all the words in the two vocabulary sets $V1$ and $V2$ have the same degree of importance. This assumption lowers the complexity of these measures. However, as will be seen in Chapter 8, this simplification may decrease the accuracy of the computed similarity value. These measures, which are quite traditional, mainly differ in their normalization factor. Due to the wide use of these measures in the domains of information retrieval and text mining, we were motivated to study their accuracy for our purposes.

  1. *Jaccard similarity*

     The well-known Jaccard similarity is one of the measures that we use in order to compute the similarity between two keywords based on their common vocabulary. Equation 7.1 shows the formula for calculating the Jaccard similarity measure.

$$Jaccard(V_1, V_2) = \frac{|V_1 \cap V_2|}{|V_1 \cup V_2|} = \frac{|V_1 \cap V_2|}{|V_1| + |V_2| - |V_1 \cap V_2|} \tag{7.1}$$

  2. *MaxDivision similarity*

     The number of common words between the two vocabularies can be also normalized by the cardinality of the longer one. The similarity measure which

applies such a normalization is called MaxDivision similarity in our work and is computed using Equation 7.2.

$$MaxDivision(V_1, V_2) = \frac{|V_1 \cap V_2|}{Max(|V_1|, |V_2|)} \tag{7.2}$$

3. *MinDivision similarity*

   In contrary to the MaxDivision, in the MinDivision similarity, the number of common words is normalized by the cardinality of the smaller set (Equation 7.3).

$$MinDivision(V_1, V_2) = \frac{|V_1 \cap V_2|}{Min(|V_1|, |V_2|)} \tag{7.3}$$

4. *AvgDivision similarity*

   In order to find the similarity between two vocabularies, AvgDivision similarity normalizes the number of common words by the average size of the vocabularies (Equation 7.4).

$$AvgDivision(V_1, V_2) = \frac{|V_1 \cap V_2|}{Avg(|V_1|, |V_2|)} \tag{7.4}$$

5. *Dice similarity*

   Dice similarity, also called Dice's coefficient, is one of the similarity measures that we tried. Equation 7.5 shows the formula for computing the similarity of the two vocabularies using Dice similarity.

$$Dice(V_1, V_2) = \frac{2 \times |V_1 \cap V_2|}{|V_1| + |V_2|} \tag{7.5}$$

6. *Cosine similarity*

   The widely-used Cosine similarity measure was tried in our approach. This measure computes the similarity between two vocabularies using the scalar product of the vocabulary vectors as normalization factor (Equation 7.6).

$$Cosine(V_1, V_2) = \frac{|V_1 \cap V_2|}{||V_1|| \times ||V_2||} \tag{7.6}$$

- *Weighted similarity measures*: These measures are based on the assumption that the words in a vocabulary do not have the same level of importance. Considering a vocabulary related to a specific domain, some words of the vocabulary contribute more to the domain of study. We exploited two different measures for assigning weights to the words. Equation 7.7 shows the general formula for calculating the weighted similarity measures, where w indicates any word of the vocabularies. These measures are traditional ones which differ in the way they compute word importance. In the following, we explain how we adapted them to corpora composed of sets of snippets.

$$\textit{Weighted similarity measure} = \frac{\sum_{w \in S_1 \cap S_2} WordImportance(w)}{\sum_{w \in S_1 \cup S_2} WordImportance(w)} \tag{7.7}$$

1. *Frequency-based similarity*

   In the Frequency-based similarity, words of a vocabulary are weighted based on their normalized frequencies in the corresponding snippet set. The assumption is that the more frequent a word is in a snippet set, the more important it is in the corresponding vocabulary. The normalized frequency of word w_i in the snippet set S is calculated using Equation 7.8.

   $$NF_{w_i,S} = \frac{Freq(w_i, S)}{N(w)} \tag{7.8}$$

   where $Freq(w_i, S)$ is the frequency of w_i in the snippet set S and $N(w)$ is the total number of words, including duplicates, in the snippet set.

   The overall importance of w_i in the two vocabularies V_1 and V_2 is computed by taking an average over the normalized frequencies of the word in the snippet sets associated to the vocabularies (Equation 7.9).

   $$WordImportance_{w_i} = \frac{NF_{w_i,S_1} + NF_{w_i,S_2}}{2} \tag{7.9}$$

2. *TF.IDF-based similarity*

   In this measure, words are weighted using the TF.IDF measure, calculated over the corresponding snippet set. Each snippet is regarded as a document and TF.IDF assigns higher weights to the words which appear frequently in one snippet but rarely in the others. After computing the weights of words in each snippet, the importance of each word in the whole vocabulary V is obtained using Equation 7.10, where $TF.IDF_{w_i,snippet_j}$ represents the weight of w_i in snippet j, calculated using TF.IDF measure. $SERP\_size$ is the number of snippets in the set.

   $$AVG\_TF.IDF_{w_i,V} = \frac{\sum_{j=1}^{SERP\_size} TF.IDF_{w_i,snippet_j}}{SERP\_size} \tag{7.10}$$

   Similar to the frequency-based measure, the overall importance of a word in the two vocabularies $V1$ and V_2 is computed by taking the average of its importance in each vocabulary (Equation 7.11).

   $$WordImportance_{w_i} = \frac{AVG\_TF.IDF_{w_i,V_1} + AVG\_TF.IDF_{w_i,V_2}}{2} \tag{7.11}$$

To sum up, the vocabulary-based similarity associates a vocabulary to each keyword and computes the similarity between the keywords through the similarity between their vocabularies. We compute this similarity using both unweighted and weighted similarity measures and in Chapter 8, we show the effectiveness of each measure.

As a future work, we are interested to try third-order similarity measures in the vocabulary-based similarity. These measures do not rely on exact matches between words

and rather use collocation measures to capture similarity. As an example, the third-order similarity measure proposed by Dias et al. (2007) could be used to study how the effectiveness of the vocabulary-based similarity would change after using a measure which does not depend on exact matches.

Co-occurrence-based similarity Co-occurrence-based similarity is proposed to capture similarity from another aspect, assuming that similar keywords are more likely to appear within the same context. Here, we aim at capturing how probable it is to find one keyword in the context of another keyword. The more probable it is, the more similar the studied keywords are.

We consider the web pages in which a keyword occurs as the context of that keyword. In concrete terms, the context of a keyword is formed by the contents of the pages returned by Google, when this search engine is queried with the keyword. Let this keyword and its corresponding pages be kw_i and $PageSet(kw_i)$, respectively. The similarity between any two keywords, kw_i and kw_j, is calculated using Equation 7.12. In more details, in order to compute how similar kw_i is to kw_j, we count the number of web pages in $PageSet(kw_i)$, where kw_j appears at least once. A keyword is considered to be in a web page if all its tokens exist at least once in the page. It should be mentioned that the contents of web pages, exploited in this step, are extracted and analyzed using the text analysis step that was introduced in Section 5.1.

$$sim(kw_i, kw_j) = \frac{\#pages\ in\ PageSet(kw_i)\ that\ contain\ all\ tokens\ of\ kw_j}{size(PageSet(kw_i))} \tag{7.12}$$

The motivation behind using page contents rather than snippets in the co-occurrence-based similarity is related to the fact that our extracted keywords can be of various length. More specific keywords mainly consist of more tokens. The probability of finding such long keywords within short snippets is low and this would lead to miss a lot of semantically related keywords. In other words, snippets do not provide enough context to capture the co-occurrence-based similarity between long keywords that we extract by our keyword extraction approach. In Chapter 8, we demonstrate this and we show that the accuracy of the measure decreases after replacing page contents by snippets.

The idea proposed by Chen et al. (2006) is close to our motivation for proposing the co-occurrence-based similarity. However, we try a different formula for capturing the co-occurrence between keywords. Furthermore, we have different complexity analysis. In their work, snippets are exploited as a source of information for capturing the co-occurrence. Although, in general, it is less complex to analyze snippets than pages, the number of retrieved snippets or page contents should be also taken into consideration as an important parameter. In other words, there should be a trade-off between the complexity of analyzing the retrieved contents and the number of requests sent to search engines. Chen et al. (2006) showed that using 600 snippets, their approach can effectively find the association between two words. We, however, did not find this number to be optimal, specially if the number of

keywords in the dataset grows. Considering the mentioned trade-off, our approach sends much fewer requests to search engines. Empirically, we found 50 pages to be enough for capturing the co-occurrence-based similarity. To reduce the complexity of analyzing page contents, we assume that having only one occurrence in a page is enough to consider it as an indication of co-occurrence between two keywords. Hence, depending on the position of the studied keyword in the web page, not all its content needs to be analyzed.

The co-occurrence-based similarity is not symmetric *per se*. Finding kw_j frequently in the context of kw_i means that while people write about kw_i, it is likely that they also write about kw_j. This co-occurrence shows the similarity of kw_j to kw_i, which is shown as $sim(kw_i, kw_j)$. The reverse relation, *i.e.* $sim(kw_j, kw_i)$, may however not be true. In other words, while someone is writing about kw_j, it might be much less likely to also discuss kw_i. As an example, *sim(cuisine, poêle)* $\neq$ *sim(poêle, cuisine)*, meaning that the probability of finding poêle in the context of cuisine is not the same as the probability of finding cuisine in the context of poêle and intuitively the latter one is higher.

Since our goal is to generate an undirected graph, asymmetric similarities need to be transformed into symmetric ones. To do this, after computing each asymmetric similarity using Equation 7.12, we tried the following ways for transforming it into a symmetric similarity.

- $AVG(sim(kw_i, kw_j), sim(kw_j, kw_i))$
- $Max(sim(kw_i, kw_j), sim(kw_j, kw_i))$
- $Min(sim(kw_i, kw_j), sim(kw_j, kw_i))$

The motivation behind trying *Minimum* is that the *Average* and *Maximum* might be biased by only one-directional high similarity value. Hence, they may consider two keywords to be similar even if they have very low similarity in another direction. The *Minimum* function, however, overcomes this limitation and considers two keywords to be similar if they have high enough similarity in both directions.

Final similarity calculation The next step is to merge the vocabulary-based and the co-occurrence-based similarities in order to obtain the final similarity value between any two keywords. In Chapter 8, we present our experiments on any of these similarities separately. We then show that exploiting them both outperforms using each of them individually.

In order to merge the two similarities, we first tried a supervised method and considered the similarities as features of a classifier. The goal was to find the weight of each feature based on a training data. As previously mentioned, generating the training data is not trivial. In addition, due to the high degree of subjectivity in determining the similarities, we obtained a low inter-agreement between evaluators. The results obtained using the supervised method were not stable enough for different kinds of test data.

As an alternative approach, we performed a series of experiments (see Chapter 8) and according to the results, we experimentally found thresholds for each similarity. We found

this alternative approach to be more stable and less complex. If both similarities are greater than their corresponding threshold values, the keywords are detected as similar and an edge is generated between them in the final graph. Hence, the final similarity is binary, which leads to generating unweighted edges. Table 7.1 shows some examples of the computed similarities. The final decision on the similarity between two keywords has been made according to the threshold values that we found empirically. These thresholds will be introduced in Chapter 8.

Table 7.1: Examples of the computed similarities using our proposed measures

keyword pair < i,j >	**Vocab-based**	**Co-occur-based**		**Final similarity**
<lld, location longue durée>	0.912	$< i, j >$	0.96	1
		$< j, i >$	0.7	
<salle de bains, robinet de douche>	0.378	$< i, j >$	0.14	1
		$< j, i >$	0.62	
<ustensiles de cuisine, poêle à bois>	0.172	$< i, j >$	0.025	0
		$< j, i >$	0	
<ustensiles de cuisine, poêle 24 cm>	0.335	$< i, j >$	0.175	1
		$< j, i >$	0.55	
<poêle à bois, poêle tefal>	0.105	$< i, j >$	0	0
		$< j, i >$	0.02	
<éclairage de noël, arbre lumineux led>	0.371	$< i, j >$	0.16	1
		$< j, i >$	0.3	

As shown on Table 7.1, our measures can correctly detect the similarity between a keyword and its abbreviation. They can also capture the similarity between keywords with no common token. Moreover, polysemies can be disambiguated. As an example, the word "poêle" does not have the same meaning in "poêle à bois", "poêle 24 cm" and "poêle tefal". As the similarity values show, our measures can effectively detect this difference.

7.1.2 Graph analysis

Once the similarity graph is generated, we detect the set of communities in the graph. These communities are considered as the topics of the collection of keywords for which the graph has been generated. In this thesis, the set of communities is referred to as a *cover*. Our definition should not be however confused with different definitions of a cover in other works (Lancichinetti et al., 2009). Since we may have ambiguous keywords, our algorithm of community detection needs to support overlapping communities. For this purpose, we use the widely-used algorithm of Clique Percolation Method (CPM) (Palla et al., 2005). Although other algorithms could be exploited in this step, we found CPM to be effective enough for our problem.

One limitation of CPM is that it fails to detect communities in a graph with few cliques. This, however, is not a problem for our approach, since the collection for which the graph is generated is not a random collection and is specific to a point of view. Although documents of the collection may be multi-topic, the majority of their contents are related to the studied point of view. This property leads to generating a dense graph in the topic detection step.

Another limitation of CPM is that if the graph contains too many cliques, the algorithm could return a single cluster containing all nodes of the graph. To avoid this problem, in the edge generation phase of our algorithm, the values of the thresholds are determined somehow not to connect all nodes of the graph and hence not to make a very dense graph. Having effective threshold values, we generate an ideal graph for CPM. We use *cfinder*[6] software in order to apply CPM on our generated graph.

Xie et al. (2013) did a study on different overlapping community detection algorithms and reported their performance on social networks. As a future work, we could perform a similar study on our specific graph in order to compare the performance of CPM with other overlapping community detection algorithms and to find the best algorithm for our problem.

As previously mentioned, the communities detected using CPM are considered as the topics of the collection. However, CPM does not return a single set of communities. Instead, it returns one cover for each clique size. The size of a clique, k, ranges from 3 to the size of the largest clique that exists in the graph. Automatically finding the best value of k, which gives the most meaningful structure over the graph, is a challenging task. One heuristic for finding the best value of k is to pick a critical k below which a giant community emerges. Although this analysis can be practical for some applications, for our specific problem we did not find a single value of k to return the best set of communities. Instead, our studies show that having a mixture of communities over different values of k gives the most meaningful cover for our problem. As a result, we proposed a heuristic-based algorithm, which generates one cover out of the mixture of the covers returned by CPM. The algorithm, named *communities selection algorithm*, depends on both the number of the missing nodes and the overlapping between the communities from one k to another one. In this algorithm, the coverage over the graph is aimed to be maximized. Algorithm 1 shows the pseudocode of this algorithm.

Having the output of CPM for different sizes of clique, the set of potential communities is initialized with the communities returned for the minimum size of the clique, *i.e.* $k = 3$. Starting from the next value of k, the covers are analyzed one after another. Let the current and the previous cover be C and $C-1$, respectively. In general, by increasing the size of a clique, some nodes which are not connected enough to the rest of the graph are discarded. While detecting the final communities, we are interested to have the maximum coverage over the graph and so to miss as little information as possible. Hence, in our algorithm, if $X\%$ of the nodes of one community from $C-1$ do not appear in any community of C, this is considered as a loss of information. To avoid this, the studied community from $C-1$ is added to the list of the final communities. Here, the value of X is determined empirically.

In some datasets, the similarity between keywords is rather high. As a result, the generated graph is mainly dense and different topics are not easily distinguishable. In this case, to separate the topics, we need to take advantage of larger cliques, since CPM detects many small cliques as adjacent and so may not distinguish all the topics. To achieve this,

[6] http://www.cfinder.org/

Algorithm 1: Communities selection algorithm

Data: CPM output
Result: Set of final communities

```
Initializing the set of potential communities with k = 3 data;
Final communities = ∅
for k = 4 to the maximum k do
    if X% of the nodes of one community from a lower value of k do not appear in
     any community of the current k then
        that community is added to the list of final communities;
    if one big community from lower k values is split into two or more smaller
     communities in the current k then
        the big community is replaced by the small ones in the set of potential
         communities;
    if one community misses one of its nodes then
        the bigger community is kept in the set of potential communities;
Add the potential communities to the final communities
```

our algorithm checks if one large community from $C-1$, is split into two or more smaller communities in C. In this case, to better distinguish the topics, the big community is replaced by the small ones in the set of potential communities. However, in some cases, only one node of a community in $C-1$ is missed in the corresponding community in C. By studying these communities, we found out that the larger community better represents its underlying topic. Hence, the missing node can be considered as the loss of information that we want to avoid in our recommendation. In this case, the larger community is kept in the list of the potential communities.

After analyzing all the values of k, the communities in the list of potential communities are added to the list of final communities in order to generate the final coarse-grained topics of the given collection.

To be clearer, we show an example of the communities selection algorithm. In this example, (com_i, C_k) is referred to as the i^{th} community of the cover C with clique size of k. Tables 7.2 to 7.4 show the communities returned for the different values of k.

Starting from $k = 3$, four communities are detected by CPM. The set of potential communities is initialized with these communities. The set of final communities is also empty at the beginning of the algorithm.

Potential communities: $(com_1, C_3), (com_2, C_3), (com_3, C_3), (com_4, C_3)$
Final communities: $\emptyset$

For the next value of k, *i.e.* $k = 4$, the communities are analyzed. According to the analysis, $X\%$ of the keywords of (com_1, C_3) do not appear in any community of C_4. As a result, (com_1, C_3) is added to the list of final communities. In addition, the analysis shows that (com_4, C_3) misses one of its keywords in C_4. Since this is considered as a loss

Table 7.2: Communities returned for clique size of 3

C_3			
Community 1	**Community 2**	**Community 3**	**Community 4**
carte	accessoire robot de cuisine	poêle	foyer et insert
carte bancaire	accessoires de cuisine	poêle 20 cm	insert
cartes de paiement	couteau de cuisine	ustensiles de cuisine	insert à bois
modes de paiement	couteaux inox	poêle en inox	poêle
paiement par carte bancaire	cuiseur vapeur	poêle antiadhésive	poêle ou insert
paiement par cb	cuisine companion		poêle à bois classique
paiement site	mallette fix class cuisine		poêle à bois
sans frais	robot cuiseur multifonction		
	robot de cuisine		
	robot multifonction		
	robots pâtissiers		
	ustensiles de cuisine		
	vente accessoire robot		
	vente accessoire robot pas cher		

Table 7.3: Communities returned for clique size of 4

C_4			
Community 1	**Community 2**	**Community 3**	**Community 4**
paiement par carte bancaire	accessoire robot de cuisine	poêle	insert
paiement par cb	accessoires de cuisine	poêle 20 cm	insert à bois
	couteau de cuisine	ustensiles de cuisine	poêle
	couteaux inox	poêle en inox	poêle ou insert
	cuiseur vapeur	poêle antiadhésive	poêle à bois classique
	cuisine companion		poêle à bois
	mallette fix class cuisine		
	robot cuiseur multifonction		
	robot de cuisine		
	robot multifonction		
	robots pâtissiers		
	ustensiles de cuisine		
	vente accessoire robot		
	vente accessoire robot pas cher		

Table 7.4: Communities returned for clique size of 5

C_5		
Community 1	**Community 2**	**Community 3**
couteau de cuisine	insert	accessoire robot de cuisine
mallette fix class cuisine	insert à bois	cuiseur vapeur
accessoires de cuisine	poêle	cuisine companion
ustensiles de cuisine	poêle ou insert	robot cuiseur multifonction
	poêle à bois classique	robot de cuisine
	poêle à bois	robot multifonction
		robots pâtissiers
		vente accessoire robot
		vente accessoire robot pas cher
		accessoires de cuisine
		ustensiles de cuisine

of information, we keep (com_4, C_3) in the list of potential communities. As a result, sets of potential and final communities are updated as below:

Potential communities: $(com_2, C_3), (com_3, C_3), (com_4, C_3)$
Final communities: (com_1, C_3)

For $k = 5$, similarly, (com_4, C_4) misses one of its keywords in C_5. As this is a loss of information for us, we keep the biggest community corresponding to this community. The biggest community is (com_4, C_3) that is kept in the list of potential communities.

Analysis also shows that (com_2, C_4) is split into (com_1, C_5) and (com_3, C_5): a big community is divided into small ones. According to our algorithm, the big community should be replaced by the small ones in the set of potential communities. As a result, the two sets are updated as below:

Potential communities: $(com_3, C_3), (com_4, C_3), (com_1, C_5), (com_3, C_5)$
Final communities: (com_1, C_3)

Since there is no higher level of k, communities in the potential set are added to the final set. The final set of communities, *i.e.* the coarse-grained topics, is then updated as below:

Final communities: $(com_1, C_3), (com_3, C_3), (com_4, C_3), (com_1, C_5), (com_3, C_5)$

To better show the functionality of the coarse-grained topic detection step, Figure 7.8 presents another example for an input keyword "gestion de flotte". As it is seen, three different topics exist in the collection of documents that Google returns as an answer to this query. Keywords within each topic are semantically consistent.

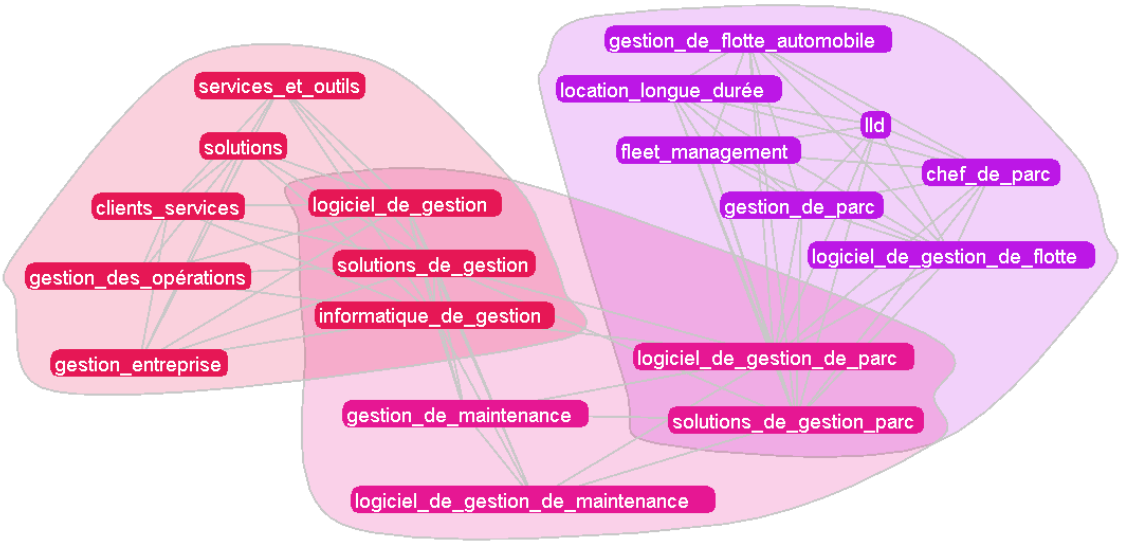


Figure 7.8: Example of the coarse-grained topic detection result

As previously explained, the coarse-grained topic detection step can manage ambiguity in case of having polysemies in the collection of keywords. An example of this disambiguation is illustrated in Figure 7.9, where different meanings of the ambiguous word "carte" have been correctly distinguished.

One property of our topic detection approach is that it is robust to the typographical errors in web pages. This property is mainly related to the co-occurrence-based similarity,

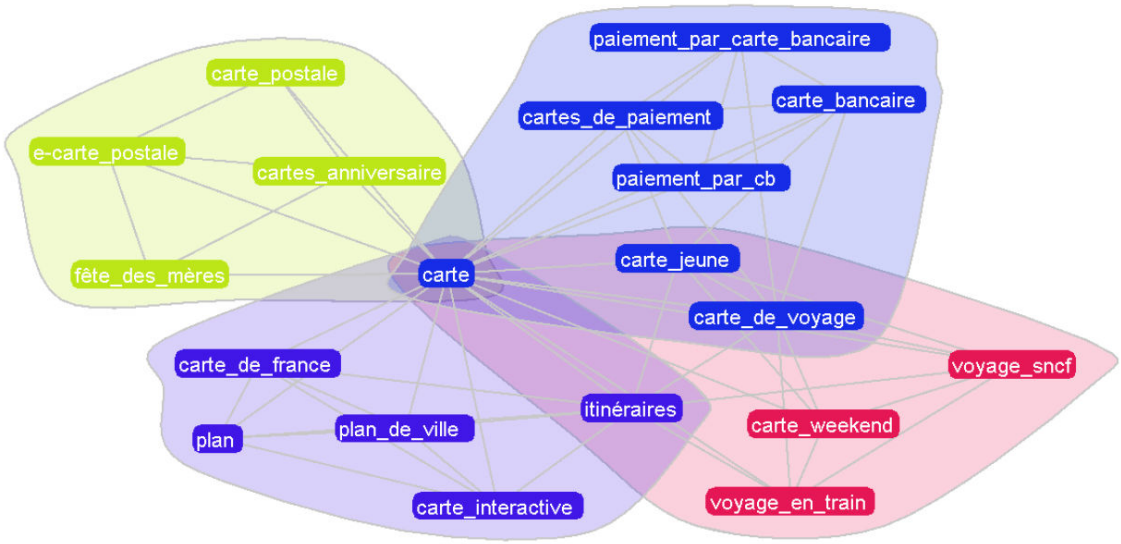


Figure 7.9: Example of the polysemy detection

which is exploited for generating edges of the graph. A keyword with a typographical error is not likely to appear frequently in different pages. As a result, the co-occurrence-based similarity cannot find it frequently co-occurrent with other studied keywords and this lowers the similarity of the keyword with other keywords. Consequently, in the generated graph, it will not be well connected to the other keywords and in the graph analysis step, it is not detected in any of the communities.

7.1.3 Selecting the relevant topics

After detecting the topics using the Clique Percolation Method (CPM) and the communities selection algorithm, the next step is to select the "relevant" ones. The relevancy of a topic is determined according to the relevancy to the enrichment's point of view that a user, who performs the enrichment process, specifies. The enrichment's point of view can be close to or far from the domain of the input document that needs to be enriched. For our specific application, we assume that the document and the enrichment's point of view belong to the same domain but our approach is robust to the case where these two discuss different domains.

Considering our assumption, a topic is relevant to the enrichment's point of view, if it contains a good ratio of the keywords of the input document. Topics which do not meet this condition but are related to the ones which contain these keywords are also considered to be relevant. Generally, we consider two topics T_i and T_j to be related if one of the following conditions holds:

- $\frac{|OverlappingNodes(T_i,T_j)|}{Min(|T_i|,|T_j|)} > Threshold_1$
- $\frac{\#\text{connections between non-overlapping nodes of } T_i \text{ and } T_j}{\#\text{all the possible connections between nodes of } T_i \text{ and } T_j} > Threshold_2$

The returned topics in this step are considered as the relevant coarse-grained topics. In the next step, each of these topics are divided into sub-topics to have a more structured information.

7.2 Fine-grained topic detection

The goal of this step is to further divide each topic into sub-topics in order to generate a set of fine-grained topics and a final recommendation, which is more structured and easier for users to interpret. In addition, the enrichment can be performed in a more specific way by adding keywords from a fine-grained topic rather than adding all keywords of a topic in a more generic way. Specially, having large and generic topics, the sub-topic division becomes essential. The fine-grained topics that we generate out of each relevant coarse-grained topic can overlap. This is to let the keywords belong to more than one topic in case that they are discussed by different topics.

As previously mentioned, the fine-grained topic detection approach is also graph-based. However, the approaches for generating and analyzing the graph are different than the ones proposed for coarse-grained topic detection. Here, two graphs are generated: one for capturing the semantic similarity between the keywords of a coarse-grained topic, and another one for making the fine-grained topics discriminative. Figure 7.10 shows the overall framework of this function and in the following, we explain each step in more details.

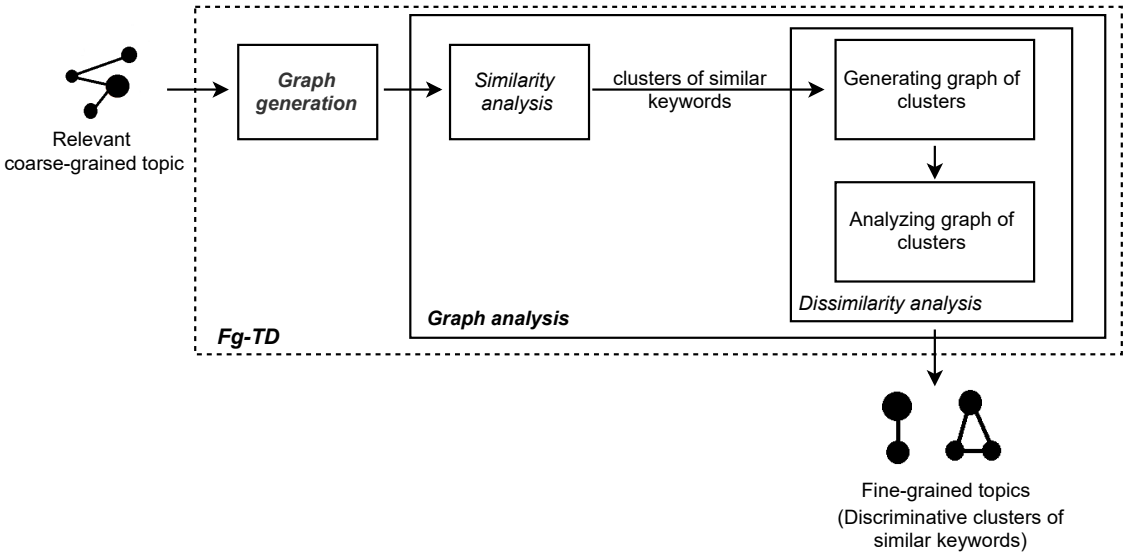


Figure 7.10: Fine-grained topic detection flowchart

7.2.1 Graph generation

As for the coarse-grained topic detection, graph generation consists of node and edge generation steps and the generated graph is undirected and weighted. The nodes of the

graph are the keywords of a target topic. Since these keywords have been already filtered in the coarse-grained topic detection step, here, we are sure about their quality and so no further processing is required.

Edges are generated in a different way. In addition to the corpus-based semantic similarity that was proposed in the coarse-grained topic detection, the morphological similarity between keywords is taken into consideration. Morphology, in general, brings good information about the similarity between keywords. However, due to the ambiguity problem, we do not exploit this information in the coarse-grained topic detection as it introduces noise. As an example, according to this type of similarity, there is a considerable degree of similarity between "carte restaurant" and "carte bancaire" due to the common word "carte", which does not have the same meaning in the two keywords. However, since disambiguation has been previously done in the coarse-grained topic detection step, this limitation is not a problem for detecting fine-grained topics. In other words, in the fine-grained topic detection step, the morphological information is reliable and can be exploited for detecting the similar keywords. Moreover, since this information relies on the common tokens, finer categorization is performed on each topic, which is an advantage for our recommendation.

As corpus-based similarity, we rely on vocabulary-based similarity. We believe that unlike the co-occurrence-based similarity, which targets semantic relatedness, vocabulary-based similarity aims at detecting semantic similarity. Although coarse-grained topics may contain both semantically similar and semantically related keywords, fine-grained topics are expected to mainly contain semantically similar keywords in order to target a more specific subject. We independently tried vocabulary-based and morphological similarities as measures for generating edges of the graph. We also tried the combination of both to see how the quality of the topics changes. Our side experiments show that using both vocabulary-based and morphological similarities gives better fine-grained topics. As a future work, it would be interesting to test the co-occurrence-based similarity and evaluate its performance in different combinations of similarity measures in order to justify our assumption about it. As morphological measure, in our approach, we rely on Jaccard similarity.

In fine-grained topic detection step, nodes of the graph are weighted and these weights are indications of nodes' importance and relevancy within the corresponding topic. We take two points into account in order to weight a node:

1. We check how important and informative the node is *per se*. To evaluate the informativeness and the importance of a node (keyword), we make use of its score, obtained in the keyword extraction step. We recall that in case of having duplicate and near duplicate keywords in the collection of keywords extracted from the enrichment collection, the weights of the unique keywords are taken from the node generation phase of the coarse-grained topic detection step (see 7.1.1). The assumption is that the more informative and important a keyword is, the higher score it will get in the keyword extraction step. In the following, we call this score the *keyword weight*.

2. We check how relevant the node is to the studied topic. To check the relevancy of a keyword in a studied topic, we make use of the concept of centrality in the graph theory in order to calculate the importance of the keyword in the graph. In graph theory, indicators of centrality identify the most important nodes within a graph. In our approach, we tried three centrality measures: *degree, betweenness,* and *closeness.* In an undirected graph, the degree centrality simply corresponds to the number of neighbors that a node has. Betweenness centrality of a node is the number of the shortest paths in the graph that pass through the node. Closeness measure corresponds to the sum of the length of the shortest paths between the node and all other nodes in the graph. All these measures capture the importance of the node within the graph from a specific point of view. We aim at identifying the centrality measure, which captures the best the relevancy property for our problem.

To take both importance and relevancy properties into consideration, their corresponding values should be aggregated. In our approach, we simply use *average* as the aggregate function. We tried three combinations for the average function: *(keyword weight & degree)*, *(keyword weight & betweenness)*, *(keyword weight & closeness)*. We empirically found out that the betweenness centrality is the best indicator of relevancy in our application. As a result, an average over the keyword weight and the betweenness value is used to rank the nodes of the graph.

It would be interesting to try more complex aggregate functions, where the two properties do not necessarily have the same level of importance. But this is left for a future work.

7.2.2 Graph analysis

The graph analysis step has two levels: analyzing the semantic similarity between keywords to generate semantically consistent clusters and analyzing dissimilarity between clusters to generate discriminative clusters, which are considered as the final fine-grained topics. Here, we explain each level in more details.

Similarity analysis

The goal of this level is to group semantically similarity keywords into different clusters. Comparing to relations in coarse-grained topics, in this step, keywords are aimed to have stronger similarities. To achieve this, we tried three different algorithms to cluster the generated graph into a set of semantically similar clusters. In the following, we list the algorithms and explain the motivation behind trying them. We note that the first and the third algorithm support overlapping nodes between clusters. This is an important feature, since it lets the keywords belong to more than one topic in case that they are discussed by different topics. In this thesis, we do not present the experiments on the choice of the most effective algorithm for our problem.

Clique Percolation Method (CPM) (Palla et al., 2005) In our approach, we performed CPM on the newly generated graph to group the keywords into semantically similar clusters. Since CPM was effective for coarse-grained topic detection, the motivation was to see its performance in this step. However, our side experiments show that CPM cannot effectively detect fine-grained topics. This low performance can be justified by the different graph generation steps and also the different expected outputs, in terms of the degree of connectivity in the detected communities, that we have in the two steps of topic detection.

Girvan-Newman algorithm (Newman and Girvan, 2004) This algorithm is based on the edge betweenness values in the graph. The assumption is that by removing edges with high betweenness values, communities within the graph are detected. Since this algorithm has been used in other graph-based approaches of topic detection (Sayyadi and Raschid, 2013, Sayyadi et al., 2009), we were interested to see its performance on our specific graph. As before, we found out that this algorithm does not return effective enough clusters. It should be noted that the Girvan-Newman algorithm, originally, does not support overlapping nodes and Sayyadi et al. (2009) apply a heuristic to support this kind of nodes. According to this heuristic, before the graph analysis step, significant edges in the graph are duplicated along with their corresponding nodes. We, however, applied the basic model of the algorithm in our approach.

Maximal clique algorithm There are different algorithms for finding maximal cliques in a graph. In our approach, we exploit the one proposed by Eppstein et al. (2010), which is the algorithm used in the *max_cliques* function of the *igraph* library[7]. We were motivated to try this algorithm, since fine-grained topics need to be as dense and as topic-centric as possible and these properties can be obtained using the maximal cliques. On the other side, since the generated graph for our problem is dense enough, representative maximal cliques can be extracted from it. Comparing to the other two algorithms, Maximal clique algorithm returns more consistent clusters. Hence, we exploit it in the similarity analysis step.

We perform a basic analysis on the extracted maximal cliques in order to eliminate the redundant ones. Here, we rather use the notion of "cluster", since after this analysis, the maximal cliques may not have this property anymore. In our work, we consider two maximal cliques to be redundant if they meet one of the following conditions:

1. if one maximal clique is a subset of another one;
2. if they share enough keywords;
3. if they have a common representative keyword, which is an indication that they are referring to the same subject.

[7] http://igraph.org/r/

Clearly, in case of having the first condition, the larger maximal clique is kept for further processing. Keeping the larger clique is to avoid missing information while merging the maximal cliques. Having both the second and the third conditions, the studied maximal cliques are merged and the resulting cluster is sorted by the rank of the constituent keywords. The clusters are then passed to the dissimilarity analysis step, where discriminative clusters are generated and returned.

Dissimilarity analysis

Fine-grained topics that are recommended to users must be discriminative to cover different subjects of the target coarse-grained topic. However, the clusters returned by the similarity analysis step may not have this property and could refer to the same subject. To eliminate such close clusters and to eventually recommend discriminative topics, we propose a heuristic-based algorithm that goes through the clusters and returns discriminative sets of keywords. If a cluster cannot be discriminated from the other clusters, it is merged with the semantically closest one. This analysis is performed using a graph-based approach that we explain in the following. We name our heuristic-based algorithm the *DFT algorithm*, which stands for the Discriminative Fine-grained Topics. The set of clusters returned by this algorithm are considered as the fine-grained topics that are recommended to users as the result of the enrichment approach.

DFT algorithm consists of two steps: *Generating graph of clusters* and *Analyzing graph of clusters*. In the following, we explain these steps in more details.

Generating graph of clusters. In DFT, nodes are the clusters passed to the dissimilarity analysis step and edges show their semantic similarity. To simplify the representation of the graph, nodes are labeled with the representative keywords of the clusters. We determine the similarity of two clusters based on the similarity of their corresponding representative keywords. Since the representative keywords highly indicate the subject of the clusters, we can use them for capturing their similarity. Here, the goal is to assure that the studied clusters do not point to the same subject. So, we found the semantic similarity to be a more important criterion than the semantic relatedness and in the edge generation step, we exploit the vocabulary-based similarity for generating edges of the graph. However, in the future, it would be interesting to study the effect of the co-occurrence based similarity on our proposed algorithm.

As an example, suppose that the six clusters in Table 7.5 are given as inputs to DFT algorithm. The keywords in each cluster are ranked based on the importance and relevancy properties. The similarity graph of the input clusters is shown in Figure 7.11. We note that a pre-defined threshold value is used to generate the edges of the graph. Our preliminary experiments on the threshold value showed that 0.65 is an effective enough value. In other words, if similarity between the representative keywords of the clusters is more than 0.65, the clusters are not discriminative enough and so are connected within the graph.

Table 7.5: Example of the input clusters in DFT algorithm

Cluster 1	**Cluster 2**	**Cluster 3**
poêles à granulés poêles mixtes bois et granulés	poêle à bois poêle à granulés de bois poêles mixtes bois et granulés type de poêle à bois installer un poêle à bois insert ou poêle à bois poêle et insert poêles mixtes acheter un poêle	poêles à pellets poêle à granulés et cheminée chauffage pellets poêles à granulés chaudière à pellets
Cluster 4	**Cluster 5**	**Cluster 6**
poêle à granulés de bois producteur de granulés bois chaudière à granulés de bois vente de granulés de bois fabrication des granulés de bois prix du granulés de bois sacs de granulés de bois palette de granulés de bois installer un poêle à granulés inserts à granulés de bois poêles mixtes bois et granulés stocker des granulés de bois poêles mixtes	bois de chauffage vente en ligne de bois de chauffage bois de chauffage pas cher prix du bois de chauffage pellets et granulés de bois pour chauffage appareils de chauffage au bois	bûche calorifique bûches de bois bûches densifiées bois et granulés bûches compressées bois densifié bois compressé bûche éco bûches longues durées

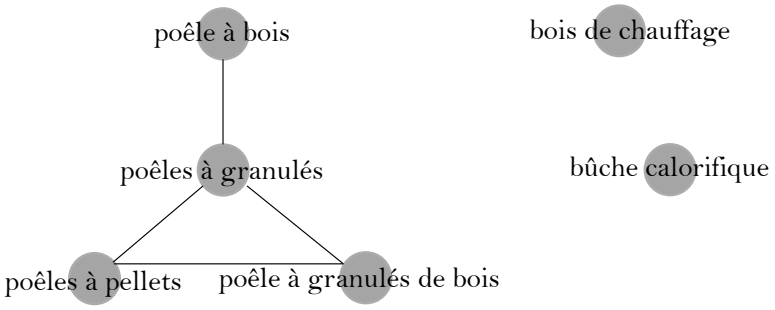


Figure 7.11: Similarity graph of the input clusters

As it is seen in Figure 7.11, the two representative keywords "bois de chauffage" and "bûche calorifique" are already disjoint. Hence, they are dissimilar enough to the other representative keywords and their corresponding clusters are discriminative enough for the final recommendation.

Analyzing graph of clusters. As previously mentioned, in the process of generating discriminative clusters, some of the input clusters may be changed into the discriminative ones. By "changing a cluster", we mean changing the order of its constituent keywords and consequently changing its representative keyword. If a cluster cannot be discriminated, it is merged with the semantically closest cluster in the graph, which obviously grows after the merge but keeps the same representative keyword.

The generated graph of clusters is globally analyzed to be transformed into a *null graph*, where there is no edge in the graph. The null graph shows that there is no significant

similarity between the representative keywords and as a result their corresponding clusters are discriminative enough to be considered as the final fine-grained topics of our approach.

Due to the complexity issues and also to avoid missing information, we are interested to have the minimum number of changes in the clusters. To achieve this, we analyze the representative keywords in order of their *betweenness* values. We recall that a node with a high betweenness value is a critical node in the graph, as the number of the shortest paths that pass through this node is high. Hence, by removing this node, it is more probable to obtain disjoint components in the graph. Since our final goal is to make the graph fully disjoint, we found betweenness as a good measure for decomposing the graph with the minimum number of changes in it. However, if the betweenness value of all the nodes is zero and the nodes are not fully disjoint, nodes will be analyzed in order of their *weights*, obtained from the keyword extraction step. In this case, the goal is to apply changes on the nodes which are less important and so have lower weights.

Hence, the algorithm starts with a node with the highest betweenness value and analyzes its corresponding cluster. The goal of this analysis is to find another representative keyword for the cluster, which is dissimilar to all other representative keywords in the graph and so to transform the cluster into a discriminative one. However, not all the keywords of a cluster are eligible to be a representative keyword. To select a keyword as a candidate representative keyword, two conditions must hold:

1. if it is an important enough keyword in the studied coarse-grained topic;
2. if it is relevant enough to the subject of the studied cluster.

We verify both conditions based on the ranking of the keyword within the cluster that was previously computed using the weight and the betweenness values (see 7.2.1). In more details, studying keywords of a cluster from the highest ranked keyword to the lowest ranked one, the ranking value of the keyword is firstly checked. In case of having a high enough rank, the keyword is considered to be important enough to be selected as a candidate representative keyword. In the next step, similarity of this candidate keyword is checked with other representative keywords in the graph. If the keyword is dissimilar to all, it is considered as a new representative keyword and its rank is updated to be the highest ranked keyword of the cluster. However, if the keyword is similar to any of the existing representative keywords, the next highly ranked keyword of the cluster is checked in turn. This iteratively continues till one of these conditions holds:

1. a new representative keyword is found;
2. no eligible keyword exists in the cluster;
3. all the keywords of the cluster are checked.

In the last two cases, the studied cluster cannot be transformed into a discriminative one as it is highly similar to one or more other clusters. Instead, we merge this cluster

with the similar cluster in the graph. In case of having more than one similar cluster, the studied cluster is merged with the cluster of the representative keyword which has the lowest *degree* value. We rely on the degree measure, since a node with a lower degree value has fewer connections to other nodes. Hence, changing it will affect less the whole graph and it also makes the algorithm converge faster. It should be also noted that if the similar representative keywords have the same degree values, we choose the one with a higher *weight*, under the assumption that the higher the rank of a representative keyword is, the more important its corresponding cluster will be. By merging the studied cluster with the cluster of the higher ranked keyword, we make a more important cluster richer, which is an advantage for our recommendation.

In the following, we explain the graph analysis step on the graph of clusters that was represented in Figure 7.11. The constituent keywords of each cluster are also presented in Table 7.5. The betweenness values of all the representative keywords are listed in Table 7.6.

Table 7.6: Betweenness values of the representative keywords

Betweenness	
poêles à granulés	2
poêles à pellets	0
bois de chauffage	0
poêles à granulés de bois	0
bûche calorifique	0
poêle à bois	0

Starting from the node with the highest betweenness value, *i.e.* "poêles à granulés", we analyze its corresponding cluster to see if any other keyword can be chosen as the representative keyword. The next keyword in the cluster is "poêles mixtes bois et granulés". Assuming that the keyword is eligible to be a candidate representative keyword, its similarity with the other representative keywords is checked. According to the vocabulary-based similarity and the similarity threshold between representative keywords, this candidate keyword is considered to be similar to two other representative keywords: "poêle à granulés de bois" and "poêle à bois". Due to this similarity, this keyword cannot be selected as a new representative keyword of the cluster. Since there is no more keyword in the cluster, Cluster 1 must be merged with one of its similar clusters. As in the graph, the degree of "poêle à bois" is lower than the one for "poêle à granulés de bois", Cluster 1 is merged with Cluster 2 and so the latter cluster is updated as in Table 7.7. The graph of the clusters is then changed accordingly (Figure 7.12).

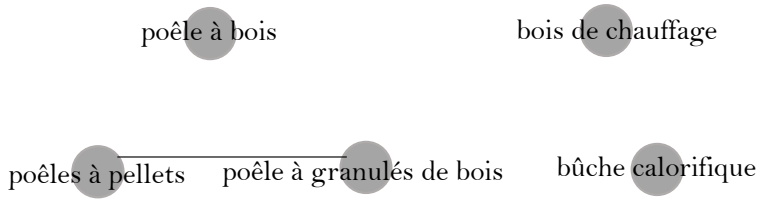


Figure 7.12: Updated graph after merging Cluster 1 and Cluster 2

Table 7.7: Updated Cluster 2 after merging with Cluster 1

Updated Cluster 2
poêle à bois
poêle à granulés de bois
poêles à granulés
poêles mixtes bois et granulés
type de poêle à bois
installer un poêle à bois
insert ou poêle à bois
poêle et insert
poêles mixtes
acheter un poêle

In the new graph, the betweenness value of all the nodes is zero but the graph is not fully disjoint yet. Hence, the next target node is selected based on the weight of the keywords. Supposing that "poêle à granulés de bois" has a higher weight than "poêles à pellets", the cluster corresponding to the latter representative keyword is analyzed. In this cluster, the second highest ranked keyword is "poêle à granulés et cheminée" that we suppose to be eligible as a candidate keyword. According to the vocabulary-based similarity and the pre-defined similarity threshold, "poêle à granulés et cheminée" is considered to be similar to "poêle à granulés de bois". Hence, this keyword cannot be a representative keyword and so the next eligible keyword of the cluster, *i.e.* "chauffage pellets", is checked. Supposing this keyword to be dissimilar enough to all the representative keywords, it becomes the new representative keyword of the cluster. As a consequence, Cluster 3 is updated as in Table 7.8.

Table 7.8: Updated Cluster 3 after finding a new representative keyword

Updated Cluster 3
chauffage pellets
poêles à pellets
poêle à granulés et cheminée
poêles à granulés
chaudière à pellets

As Figure 7.13 shows, after updating Cluster 3, all the nodes of the graph are disjoint and so the algorithm stops and returns a set of discriminative clusters, called fine-grained topics. These topics are then recommended to the user to be later added to the content of the input document.

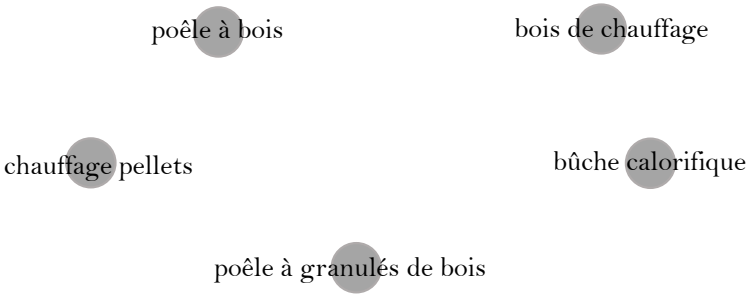


Figure 7.13: Disjoint graph of clusters (null graph)

To better show the functionality of the fine-grained topic detection, we performed it on a coarse-grained topic that was initially detected for "protection auditive" domain. Table 7.9 demonstrates the initial coarse-grained topic as well as the two overlapping fine-grained topics obtained after performing the fine-grained topic detection. Keywords in the fine-grained topics are ranked based on their importance and relevancy to the main subject of the topic. According to the representative keywords, "protections anti bruit" and "protection auditive" are two of the main subjects in the domain of study that can be separately recommended to the user for the enrichment purpose.

Table 7.9: Example of fine-grained topic detection result

Coarse-grained topic	**Fine-grained topic 1**
protections anti bruit	protections anti bruit
bouchon anti bruit	bouchon anti bruit
casque antibruit atténuation sonore	casque antibruit atténuation sonore
bouchon d'oreille anti bruit	bouchon d'oreille anti bruit
protection auditive	protection auditive
protection auditive chasse	**Fine-grained topic 2**
bouchons de protection	protection auditive
protections auditives standards	protection auditive chasse
protection auditive sur mesure	protections auditives standards
protecteur auditif	protection auditive sur mesure
oreille de protection	protection auditive casque
protection auditive casque	protections auditives musique
bouchon oreille sur mesure	protection auditive mousse confort
protections auditives musique	protection auditive concert
protection auditive mousse confort	bouchons de protection
protection auditive concert	protecteur auditif
	oreille de protection
	bouchon oreille sur mesure
	protections anti bruit

Chapter 8

Topic Detection Evaluation

Contents

In this chapter, we present our experiments on the topic detection approach and show the results that we obtained. There are different steps that we aim to evaluate and we did individual experiments and evaluations on each in order to justify our choices. The experiments are performed on multi-domain datasets to study the robustness of our approach over different domains. In general, we follow two evaluation strategies by performing a user-based evaluation and by comparing our approach with a baseline one. The first strategy shows the effectiveness of our approach when applied on real case studies of recommendation, while the second one aims at comparing its performance with respect to the state of the art.

There are several steps in our topic detection approach, where we choose an effective algorithm or pick an optimal threshold value. Since a comprehensive evaluation of all these steps would be very demanding for users, we focus on the main functions of the approach, which highly affect its performance and are related to our contribution to this work. More specifically, we evaluate four points: 1) the proposed similarity measures in the edge generation step, 2) the performance on filtering out the keywords which do not belong to the target point of view, 3) the representativeness of the representative keywords in the fine-grained topics, and 4) the semantic-consistency of the detected fine-grained topics. The user-based evaluation is performed on each point as well as on the overall approach of topic detection, which is compared with a baseline approach. Other choices of the algorithm have been fixed through side experiments that we do not present in this thesis.

Our topic detection approach was initially implemented for French. However, as discussed before, it is easily tunable to other languages. For a new language, we need to adapt the list of stop words and also the online dictionary used in the brand detection step

(see 7.1.1). A language-specific tagger is also required. Currently, we have implemented and tested the approach on English as well. However, for the evaluation *per se*, we target the French language, which is the native language of our evaluators.

In the following sections, we present our experiments and show the obtained results along with their evaluation. The chapter ends with a conclusion over the findings.

8.1 Similarity measures

In order to evaluate the effectiveness of our proposed similarity measures in capturing semantic similarity and semantic relatedness between keywords, we perform a user-based evaluation. For this purpose, a gold standard set is generated manually, which consists of both similar and dissimilar keywords. The accuracy of each similarity measure is then computed based on the match between its generated results and the gold standard set. In the following, we explain our experimental data and present the results obtained using the similarity measures.

8.1.1 Experimental data

The experimental data for evaluating the similarity measures is a manually generated gold standard set consisting of pairs of keywords. To study the robustness of the measures over different domains, the set contains 1000 pairs from 20 diverse domains. Table 8.1 summarizes these domains. In this table, each domain is specified with a keyword, which is a representative keyword of that domain. The gold standard set is designed by three evaluators, who are French native speakers. The evaluators are expert in the target application and familiar with the target domains. We pass the pairs of keywords to the evaluators and ask them to specify whether they are rather similar or rather dissimilar.

Since determining the similarity is a subjective task and is different from one evaluator to another, we make use of Fleiss' kappa statistics (Fleiss et al., 1971), which enables us to compute the inter-agreement between the three evaluators. Over the whole gold standard set, we obtained 0.81 as the inter-agreement between the evaluators. The Fleiss' kappa (k) is computed using Equations 8.1 and it gives a measure for how consistent the ratings are. In this equation, $1 - \overline{P}_e$ is the degree of agreement that is attainable above chance and $\overline{P} - \overline{P}_e$ is the degree of agreement that actually achieved above chance. $\overline{P}$ is the mean of the extent to which raters agree for each entry and is computed using Equation 8.2, where N is the total number of entries, n is the number of the ratings per entry, K is the number of categories into which assignments are made and n_{ij} shows the number of raters who assigned the i^{th} entry to the j^{th} category. $\overline{P}_e$ is computed using Equation 8.3, where P_j is the proportion of all assignments which were to the j^{th} category (Equation 8.4). For our evaluation, values of N, K and n are respectively set to 1000, 2 and 3.

$$k = \frac{\overline{P} - \overline{P}_e}{1 - \overline{P}_e} \tag{8.1}$$

Table 8.1: Domains in the gold standard set

	Domain	Number of pairs of keywords
1	abri de jardin	50
2	agence seo	50
3	assurance maison	50
4	bois de chauffage	50
5	brosse soufflante	50
6	certification	50
7	collier de serrage	50
8	envoi colis	50
9	étude économique	50
10	gaz naturel	50
11	isolation	50
12	lasure	50
13	lit électrique	50
14	lustre et suspension	50
15	mobilier de salle de bain	50
16	mozzarella	50
17	outillage électroportatif	50
18	peinture cuisine	50
19	protection auditive	50
20	tableaux électriques	50

$$\overline{P} = \frac{1}{Nn(n-1)}(\sum_{i=1}^{N}\sum_{j=1}^{K} n_{ij}^2 - Nn) \tag{8.2}$$

$$\overline{P}_e = \sum_{j=1}^{K} P_j^2 \tag{8.3}$$

$$P_j = \frac{1}{Nn}\sum_{i=1}^{N} n_{ij} \tag{8.4}$$

According to Landis and Koch (1977), Fleiss' kappa value between 0.81 and 1 indicates an almost perfect agreement. Hence, we consider that the agreement between the evaluators is good enough for the similarity evaluation. To make the final decision on the similarity between keywords, we use the majority vote across the evaluators' decisions. Eventually, our gold standard set contains 437 similar and 563 dissimilar pairs of keywords. Table 8.2 shows examples of the pairs in the gold standard set along with their similarity labels.

8.1.2 Experimental results and evaluation

The vocabulary-based and the co-occurrence-based similarities are firstly evaluated individually. For each type of similarity, the best measure/function for computing the similarity values and the best threshold for modeling the connectivity in the edge generation step are

Table 8.2: Examples of the evaluated pairs in the gold standard set

Keyword 1	**Keyword 2**	**Similarity label**
isolation sonore	isolation phonique	1
isolation sonore	site internet	0
panneaux mdf	bois	1
appareil auditif	anti bruit	0
casque anti bruit electronique	achat en ligne	0
agence web	search marketing	1
peinture de cuisine	cuisine en bois	0

empirically found. We recall that in the vocabulary-based similarity, we try 8 measures to compute the similarity between the vocabularies associated to two keywords. In the co-occurrence-based similarity 3 different functions are also tried to make the similarity symmetric and applicable on an undirected graph. For each case, the optimal threshold value, above which pairs of keywords are considered similar, is different. We perform an experiment over each case individually. The similarities obtained by each measure/function are compared with the gold standard ones and accordingly, the best measure for the vocabulary-based similarity and the best function for the co-occurrence-based similarity along with their optimal threshold values are found. The similarities are evaluated according to their *accuracy*, which is defined as the ratio of the match between the automatic and the manual similarity labels.

In all the experiments, for each keyword, 50 web pages or snippets in the search engine result page (SERP) are exploited as the context of the keyword. We initially started with 40 pages/snippets. The number was then increased to 50, which led to slightly better results. Since 50 number of pages/snippets already gave acceptable results for our problem, we did not increase more the size of the SERP context in order to control the complexity of our approach. The experiments that led to fix the size of SERP context to 50 pages/snippets are not reported here.

Vocabulary-based similarity

We recall that in the vocabulary-based measures, snippets of the SERP context are used as a source of information. In addition, a threshold value specifies the value above which keywords are considered to be semantically similar/related. In order to study the accuracy of different measures for the vocabulary-based similarity, we perform the same experiment over all the measures. For each one, different threshold values are tested, starting from 0.05 as the minimum similarity threshold between two keywords to 0.5. The results are then compared with the gold standard set and the accuracy of each measure on a specific threshold is computed. Table 8.3 shows the result of these experiments.

As mentioned before, the vocabulary-based similarity has been inspired by the similarity measure proposed by Sahami and Heilman (2006), who expand the vocabulary of each keyword using pages contents. We did not get outstanding results when performing

Table 8.3: Experiments on the 8 measures of the vocabulary-based similarity with snippets as context

Jaccard										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	0.474	**0.725**	0.707	0.624	0.586	0.574	0.572	0.569	0.567	0.567
MaxDivision										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	0.439	0.506	0.679	**0.749**	0.69	0.633	0.605	0.579	0.574	0.572
MinDivision										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	0.437	0.459	0.577	0.713	**0.746**	0.691	0.626	0.595	0.583	0.576
AvgDivision										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	0.438	0.48	0.633	**0.755**	0.718	0.653	0.612	0.586	0.575	0.572
Dice										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	0.438	0.48	0.634	**0.755**	0.719	0.654	0.612	0.586	0.575	0.572
Cosine										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	0.438	0.481	0.627	**0.757**	0.72	0.655	0.612	0.586	0.575	0.572
Frequency-based										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	0.45	0.563	0.692	0.767	**0.777**	0.755	0.713	0.653	0.622	0.6
TF.IDF-based										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	0.458	0.648	**0.759**	0.72	0.639	0.605	0.585	0.576	0.572	0.57

preliminary experiments on their measure. Our intuition is that this could be due to the weighting schema and the source of information that they exploit in their work. As a consequence, in the following, we specifically focus on these two criteria and discuss their effectiveness on the vocabulary-based similarity. Since the details of the two similarity measures are different, it would be interesting to present formal experiments on the original measure of Sahami and Heilman (2006) and to compare it with the vocabulary-based similarity. Nevertheless, in this thesis, we only rely on our preliminary experiments and do not perform a formal comparison between the two measures.

According to Table 8.3, in the vocabulary-based similarity, the highest value of accuracy, *i.e.* 0.777, corresponds to the Frequency-based measure, which differs from the results of Sahami and Heilman (2006), who use TF.IDF as a vector weighting scheme.

In addition to the overall accuracy computed over all the 1000 pairs, we perform experiments on each of the 20 domains in order to study the effectiveness of each measure in each domain and the robustness of the measures across different domains. Table 8.4 illustrates the number of the domains for which each measure performs the best. According to the results, the Frequency-based measure is the best measure on 12 domains and this confirms its robustness.

To compare our weighting measure with the one used by Sahami and Heilman (2006), we specifically study the standard deviation of the Frequency-based and the TF.IDF-

Table 8.4: The number of domains for which each measure performs the best. For some domains more than one measure performs the best. Hence, the sum over the number of domains exceeds the total number of domains, *i.e.* 20.

Measure	Number of domains
Jaccard	1
MaxDivision	1
MinDivision	5
AvgDivision	1
Dice	1
Cosine	2
Frequency-based	12
TF.IDF-based	5

based measures. These values are presented in Figure 8.1, where plot of the Frequency-based measure is variating less, indicating that it generates more stable results. Since our enrichment approach is expected to be domain-independent, robustness of the measure is highly important.

Considering both accuracy and robustness, the Frequency-based measure is the most effective measure of vocabulary-based similarity.

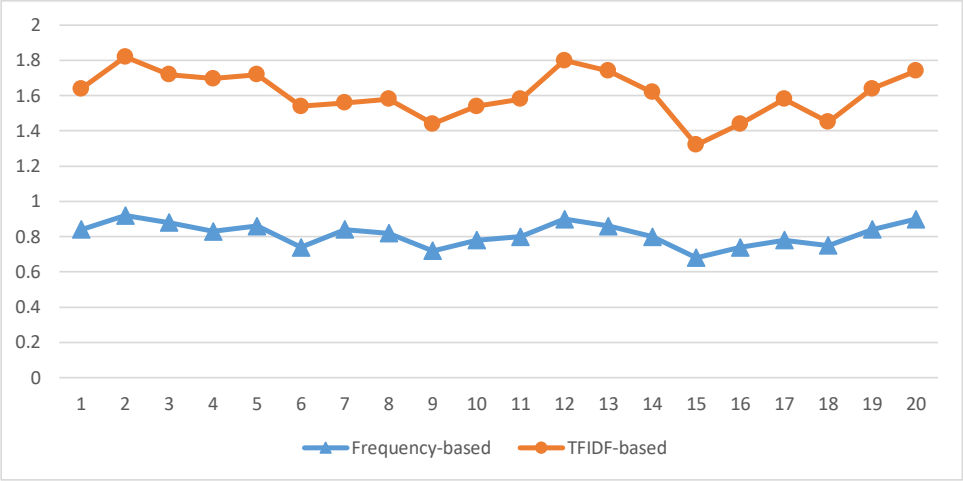


Figure 8.1: Standard deviation of the Frequency-based and the TF.IDF-based measures across the 20 domains

Unlike our work, which uses snippets as a source of information in the vocabulary-based similarity, Sahami and Heilman (2006) exploit page contents to expand the vocabulary of a keyword. In order to evaluate the effectiveness of snippets with respect to page contents, we perform a set of experiments, where content of web pages are used as a source of information in the vocabulary-based similarity. Table 8.5 presents the results obtained for different measures and various threshold values. In these experiments, we increase the threshold value up to 0.55, as we found higher values of thresholds to be optimal for this source of information.

Table 8.5: Experiments on the 8 measures of the vocabulary-based similarity with page contents as context

Jaccard											
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55
Accuracy	0.437	0.437	0.44	0.479	0.547	0.547	**0.576**	0.574	0.572	0.571	0.565
MaxDivision											
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55
Accuracy	0.437	0.437	0.437	0.437	0.448	0.479	0.518	0.571	0.567	**0.575**	0.574
MinDivision											
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55
Accuracy	0.437	0.437	0.437	0.437	0.437	0.442	0.452	0.505	0.548	**0.581**	0.591
AvgDivision											
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55
Accuracy	0.437	0.437	0.437	0.437	0.437	0.454	0.489	0.547	0.571	**0.592**	0.577
Dice											
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55
Accuracy	0.437	0.437	0.437	0.437	0.437	0.454	0.489	0.547	0.571	**0.592**	0.577
Cosine											
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55
Accuracy	0.437	0.437	0.437	0.437	0.437	0.449	0.486	0.546	0.572	**0.593**	0.575
Frequency-based											
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55
Accuracy	0.437	0.437	0.437	0.437	0.438	0.446	0.459	0.492	0.505	**0.535**	0.57
TF.IDF-based											
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55
Accuracy	0.437	0.437	0.437	0.437	0.438	0.477	0.519	0.573	**0.585**	**0.585**	0.574

As Table 8.5 shows, Cosine is reported to be the best measure for computing the vocabulary-based similarity using page contents. However, the maximum value of accuracy obtained using page contents is considerably lower than the one for snippets (0.777 *vs.* 0.593). Therefore, we conclude that snippets are a more effective source of information for the vocabulary-based similarity. In addition, analyzing snippets is less complex compared to page contents. Hence, this source of information both achieves a higher accuracy and lowers the complexity of the measure.

Co-occurrence-based similarity

We perform the same set of experiments on the co-occurrence-based similarity. As explained before, this similarity is not symmetric *per se* and we tried three functions to make it symmetric and applicable on an undirected graph. Similar to the evaluation of the vocabulary-based measures, we study different threshold values for determining the similarity between two keywords. We recall that in co-occurrence-based similarity, page contents are used as a source of information. Table 8.6 presents the results of these experiments. The threshold values range from 0.05 to 0.5.

According to Table 8.6, the *Average* aggregation function returns the highest accuracy value. In contrast, the *Maximum* function returns the lowest value of accuracy. This can be explained by the fact that *Maximum* biases the co-occurrence-based to one-directional

Table 8.6: Experiments on the 3 functions of the co-occurrence-based similarity with page contents as context

$Average(sim_i, sim_j)$										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	0.65	0.723	0.751	**0.76**	0.743	0.708	0.678	0.656	0.641	0.624
$Minimum(sim_i, sim_j)$										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	**0.751**	0.744	0.702	0.682	0.644	0.62	0.602	0.599	0.592	0.587
$Maximum(sim_i, sim_j)$										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	0.595	0.665	0.713	0.726	**0.741**	0.732	0.72	0.712	0.698	0.68

similarity relations between keywords, which may not necessarily indicate semantic relatedness between them. As an example, "paiement" may appear frequently in pages returned for "collection de 4 verres", but the reverse occurrence is very unlikely to happen. In this example, the *Maximum* function wrongly considers the two keywords to be semantically related. We note that the *Average* function could also have this limitation but it is much less probable than in the *Maximum* function. On the other side, *Minimum* is a rather strict function, as it detects similarity between two keywords if they are similar enough in both directions. Hence, the *Average* function is not as biased as the *Maximum* function and not as strict as the *Minimum* one. Our results show that choosing an effective value of threshold for *Average* reduces its bias and makes it the best aggregation function for the co-occurrence-based similarity.

In Chapter 7, we explained that the co-occurrence-based similarity is close to the measure proposed by Chen et al. (2006), which exploits snippets as a source of information. Although the details of our similarity is different than theirs, to compare the effectiveness of page contents with respect to snippets, we perform the same experiments but using snippets as a source of information. Table 8.7 shows the results of these experiments. Using this source of information, the *Maximum* function returns the highest accuracy value. This could be due to the fact that snippets contain focused and relevant information about the keywords. Hence, even one directional similarity could be a good indicator of semantic relatedness. In other words, it is rare to have semantically unrelated keywords which co-occur in snippets and this makes the *Maximum* function effective.

Comparing the accuracy values obtained using snippets and page contents shows that snippets are not as effective as page contents in capturing the semantic relatedness in the co-occurrence-based similarity. We justify this conclusion by noting that snippets are short pieces of information, whereas our keywords could be long in terms of the number of constituent tokens. Basically, the probability of finding co-occurrent long keywords in short contents, such as snippets, is not very high and consequently, semantic-relatedness cannot be captured effectively.

Table 8.7: Experiments on the 3 functions of the co-occurrence-based similarity with snippets as context

$Average(sim_i, sim_j)$										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	**0.676**	0.637	0.623	0.609	0.602	0.601	0.598	0.591	0.576	0.57
$Minimum(sim_i, sim_j)$										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	**0.6**	0.588	0.576	0.574	0.572	0.572	0.568	0.564	0.563	0.563
$Maximum(sim_i, sim_j)$										
Threshold	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	**0.7**	0.668	0.644	0.633	0.626	0.618	0.608	0.602	0.598	0.596

Merging both similarities

In our enrichment application, we are interested in capturing both semantic similarity and semantic relatedness. Our assumption is that the former property is obtained using the vocabulary-based similarity, whereas the latter is captured using the co-occurrence-based one. To take both properties into account, we aggregate the two similarities and obtain the final similarity, which is used for generating edges in the graph generation step of the topic detection approach. To merge the similarities, we make use of the results obtained in the previous experiments. The best measure/function along with their optimal threshold values are used for determining the final similarity. Two keywords are considered to be similar/related if they are detected as similar/related keywords by both similarities. According to Tables 8.3 and 8.6, the optimal threshold values for the vocabulary-based and the co-occurrence-based similarities are 0.25 and 0.2, respectively. Table 8.8 summarizes the best accuracy values obtained using each similarity individually and using their aggregation.

According to the results, combining the similarities captures the semantic similarity/relatedness more effectively. Hence, unlike Sahami and Heilman (2006) and Chen et al. (2006), who focus on either semantic similarity or semantic relatedness, we take both into consideration and show an improvement over the accuracy value.

It should be mentioned that to merge the similarities, we initially tried a supervised approach to learn the optimal weight of each similarity. We, however, did not find the result to be robust over various domains and we do not present this preliminary experiment in this thesis.

Table 8.8: Comparison over the similarity measures

Similarity measure	**The best accuracy**
Vocabulary-based (snippet)	0.777
Co-occurrence-based (page)	0.76
Vocabulary-based (snippet) + Co-occurrence-based (page)	**0.813**

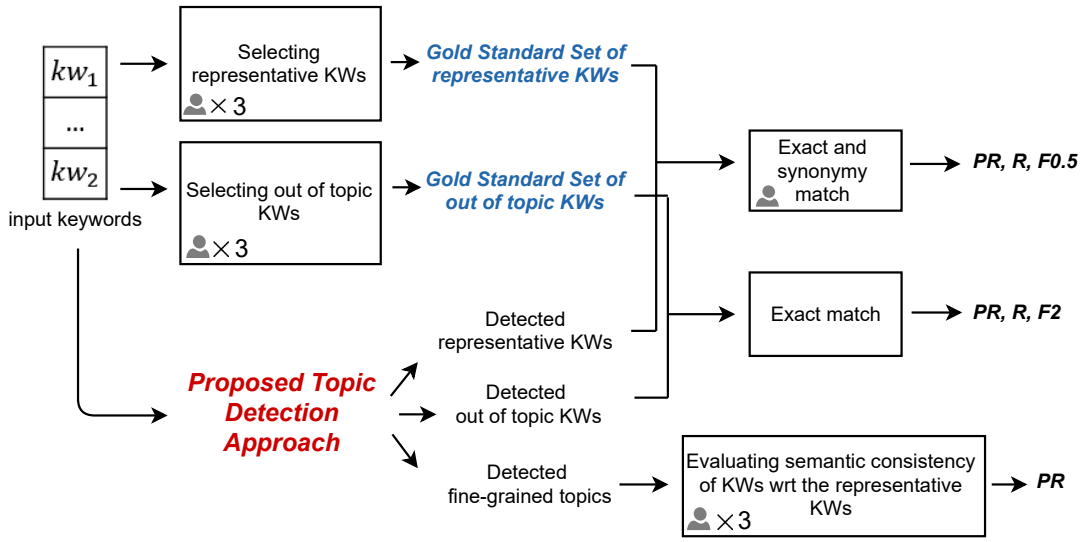


Figure 8.2: Schema of the evaluation process of coarse-grained and fine-grained topics

8.2 Recommended coarse-grained and fine-grained topics

In our approach, there are several steps for detecting the final set of coarse-grained and fine-grained topics. We evaluate the outputs that are critical for our enrichment problem: 1) out of topic keywords, 2) representative keywords, and 3) fine-grained topics. Figure 8.2 illustrates the schema of the evaluation process performed in this section. Each step of the evaluation is explained in the following.

Out of topic keywords are detected in the coarse-grained topic detection step. By out of topic keywords, we refer to the keywords that are not relevant to the target point of view. The goal of this experiment is to study the coarse-grained topic detection step and to evaluate how effective it is in filtering out the irrelevant keywords. By this evaluation, we implicitly evaluate the performance of the Clique Percolation Method (CPM) and the communities selection algorithm, explained in 7.1.2.

Representative keywords of the fine-grained topics play an important role in the enrichment application, since they specify the main subject of each topic. Focusing on these keywords, users can decide whether topics are interesting enough to be added to the input document, which needs to be enriched. If a user does not find the representative keyword of a topic interesting for the enrichment, he/she can discard the whole topic without browsing all its keywords. Hence, representative keywords make the user validation easier and faster, which is an advantage for a recommendation approach. To avoid missing information in the user validation step, representative keywords must be correctly selected and assigned to each topic. Topics recommended to the user are expected to be unique, despite of the fact that they can share keywords. In other words, the main subject of each topic must not be covered by any other topic. This requires the representative keywords to be dissimilar enough, so that each one brings a unique and discriminative piece of information.

Due to the high importance of the representative keywords, we evaluate them in this work. By studying representative keywords, we implicitly evaluate the measures for ranking keywords in the fine-grained topic detection step and also the Discriminative Fine-grained Topics (DFT) algorithm, respectively explained in 7.2.1 and 7.2.2. As ranking measures, we tried three combinations of measures: *(keyword weight & degree)*, *(keyword weight & betweenness)*, and *(keyword weight & closeness)*. This side experiment, which is not presented in this thesis, showed that the *(keyword weight & betweenness)* measure gives the best ranking over the keywords of each topic. Hence, in the final evaluation of the representative keywords, we only use this measure without asking the evaluators to study the representative keywords returned for each ranking measure.

Fine-grained topics are the final detected topics that we recommend to users. Keywords within each topic are expected to be semantically consistent. In order to show the effectiveness of the maximal clique algorithm and the heuristic-based approach that we proposed for returning the final set of fine-grained topics (see 7.2.2), we manually evaluate the semantic consistency of the returned fine-grained topics. We performed side experiments on the performance of the maximal clique algorithm with respect to CPM and Girvan-Newman algorithm but here we only focus on the best performing algorithm, *i.e.* the maximal clique.

We note that a practical evaluation would assist in generating a gold standard set of topics and comparing them with the ones detected by our approach. Generating such a gold standard set is, however, very challenging due to the subjectivity issues. The task is becoming more challenging when the number of keywords, for which the topics need to be detected, increases. For small examples, we preliminary generated gold standard sets and used the OmegaIndex measure to compare the detected topics with the gold standard ones. However, in real applications and consequently in the experiments presented in this section, the average size of datasets in terms of the number of keywords is 185 and we could not generate effective gold standard sets for these datasets. Alternatively, we focus on three outputs of our approach rather than the whole topics. As Figure 8.2 shows, evaluation of the out of topic and the representative keywords is performed by comparing the results with gold standard sets. Fine-grained topics are, however, passed to evaluators to study the semantic consistency of the keywords within each topic with respect to its representative keyword.

Although TDT benchmarks have been widely-used for the topic detection task, we do not exploit them and rather generate our own experimental data. In this experiment, we are mainly interested in evaluating the topic detection approach on the keywords extracted using our keyword extraction approach. The keyword extraction approach has been specifically proposed for web pages and may not effectively extract keywords from news stories of TDT benchmarks. This would lower the quality of the detected topics. An alternative way is to directly pass the keywords of the news stories to our topic detection approach. These keywords are however not available and an approach of keyword extraction is required to extract them. As a future work, it would be interesting to extract the keywords

of a TDT benchmark using another approach and to evaluate our topic detection approach on other types of keywords. As in Sayyadi and Raschid (2013), we could perform a very basic approach for extracting keywords of news stories.

In the following, we firstly present our experimental data. We then explain in more details the experiments on each output.

8.2.1 Experimental data

We have already executed the topic detection approach on roughly 80 domains but we target only ten domains for the purpose of evaluation.

To have a user-based evaluation on out of topic and representative keywords, we generate gold standard sets on 10 diverse domains. For each domain, 20 documents are targeted as enrichment collection and their keywords are extracted using the keyword extraction approach. All the weighted keywords are then passed to three evaluators[1]. We recall that according to the Fleiss' kappa statistics, the inter-agreement between the evaluators is 0.81. For each target domain, the evaluators analyze the proposed keywords and generate two gold standard sets, where the majority vote of the evaluators is used:

- Gold standard set of out of topic keywords: the evaluators are asked to determine the keywords which are not relevant to the target domain. As an example, let's consider the "veste" domain, in which the evaluators label "retour gratuits" as an irrelevant keyword, since it is not semantically related to the domain. We use this set for evaluating the effectiveness of our coarse-grained topic detection approach in detecting out of topic keywords.

- Gold standard set of representative keywords: the evaluators go through all the keywords of each domain individually and specify those which can be considered as representative keywords. A keyword is a representative keyword if it is targeting a specific subject and if it is discriminative enough comparing to other representative keywords. Since more than one keyword may discuss a subject, only the most common one is chosen as the representative keyword in order to avoid duplicates. As an example, having "veste" as the enrichment point of view, "veste en cuir", "veste en jean" and "veste d'hiver" can be all selected as representative keywords, as they specifically focus on special categories of "veste". On the contrary, having "veste en cuir" and "veste en cuir noir", the latter keyword is not selected as a representative keyword because it is not discriminative enough and it does not bring in any information compared to the former one.

Table 8.9 summarizes the generated gold standard sets, including the number of the input keywords in each domain along with the number of the out of topic and the representative keywords that are given by the evaluators.

[1] The same evaluators as in the evaluation of the similarity measures

Table 8.9: Statistics on the gold standard sets generated for evaluating the detected coarse-grained and fine-grained topics

Domain	#KWs	#out of topic KWs	#representative KWs
Abri de jardin	215	84	15
Isolation	182	21	35
Robinetterie	132	28	22
Assurance automobile	177	43	14
Certification	97	34	22
Collier de serrage	188	57	14
Pâtes	197	70	22
Peinture murale	275	93	24
Protection auditive	163	21	16
Tableaux électriques	225	39	15

To evaluate the semantic consistency of the fine-grained topics, we run the topic detection approach on the same sets of keywords and pass the detected topics to each evaluator. The goal is to evaluate the semantic consistency of keywords in each topic with respect to its representative keyword. Hence, we only study the topics for which the representative keywords exist in the gold standard set.

8.2.2 Experimental results and evaluation

We pass the same input keywords as in Table 8.9 to the proposed topic detection approach and evaluate its different outputs. The effectiveness of the coarse-grained topic detection step in detecting irrelevant keywords is evaluated by comparing its detected out of topic keywords with the ones labeled by the evaluators. The effectiveness is reported as *precision* and *recall*. To consider the balance between these two measures, we also compute *F-measure*. Table 8.10 shows the result of this evaluation over the 10 domains. Here, we compute F_2 measure, since recall matters more than precision. In other words, in our enrichment application, missing part of the enrichment information is acceptable but recommending wrong information, which is not related to the target point of view, is not allowed. The average F_2 measure over all the domains is 85.3%.

Table 8.10: Evaluation of the coarse-grained topic detection approach in detecting out of topic keywords

Domain	*Precision*	*Recall*	F_2
Abri de jardin	0.86	0.83	0.83
Isolation	0.64	1	0.89
Robinetterie	0.61	0.96	0.86
Assurance automobile	0.71	0.82	0.79
Certification	0.73	0.91	0.86
Collier de serrage	0.63	1	0.89
Pâtes	0.68	0.83	0.79
Peinture murale	0.66	0.95	0.87
Protection auditive	0.62	1	0.89
Tableaux électriques	0.63	0.95	0.86
AVERAGE	0.677	0.925	0.853

One interesting finding after evaluating the out of topic keywords is that our coarse-grained topic detection approach is able to mostly filter out the remaining brands in the collection of documents. Although in the graph generation step of the coarse-grained topic detection we perform a brand removal process on the input keywords (see 7.1.1), it is slightly probable that the final keywords contain brand names. We previously mentioned that our approach of brand detection is heuristic-based and may not detect all the existing brands. Nevertheless, the remaining brands can be mostly detected afterwards in the graph analysis of the coarse-grained topic detection step. This is due to the fact that brand names are basically too specific and share little vocabulary with other keywords of the domain. In addition, they do not co-occur frequently with other keywords. Hence, in the generated graph, they have a weak connectivity with other nodes and cannot be detected in the communities which are related to the domain of study.

In the next experiment, we study the representativeness of the recommended representative keywords. By this experiment, we implicitly evaluate our keywords' ranking measure and the DFT algorithm, which returns dissimilar representative keywords and so discriminative topics. For this purpose, the representative keywords detected by our approach are compared with the ones in the gold stanard set of representative keywords and the precision and recall values are computed accordingly. In this step, precision matters more than recall: we slightly accept to miss recommending a representative keyword but proposing an unimportant keyword as a representative one is not acceptable. Therefore, we compute the value of $F_{0.5}$ to emphasize more on the precision value. Table 8.11 summarizes the values of the evaluation measures obtained for each domain. The average $F_{0.5}$ over all the 10 domains is 70.5%.

Table 8.11: Evaluation of the representativeness of the recommended representative keywords

Domain	*Precision*	*Recall*	$F_{0.5}$
Abri de jardin	0.75	0.8	0.75
Isolation	0.84	0.6	0.77
Robinetterie	0.81	0.59	0.75
Assurance automobile	0.77	0.5	0.7
Certification	1	0.41	0.77
Collier de serrage	0.85	0.42	0.71
Pâtes	0.77	0.63	0.74
Peinture murale	0.66	0.66	0.66
Protection auditive	0.6	0.37	0.53
Tableaux électriques	0.69	0.6	0.67
AVERAGE	0.774	0.558	0.705

In the evaluation of the representative keywords, we should consider that two keywords could have a very close meaning. In this case, the evaluators may select one keyword as a representative keyword, while our approach may specify the other keyword as a representative one. As an example, suppose that "isolation phonique" and "isolation acoustique" exist in the dataset and that our approach returns "isolation acoustique" as the representative keyword, whereas "isolation phonique" is selected by the evaluators. Since these keywords

have a very close meaning and are mostly used interchangeably, we do not penalize our approach due to the mismatch between "isolation phonique" and "isolation acoustique". Instead, we consider them as the same keywords, while matching our representative keywords with the gold standard ones. This verification is, however, not performed automatically in our evaluation and the evaluators are asked to detect such similar keywords in the results.

We also evaluate the semantic consistency of the detected fine-grained topics. The goal of this step is to measure how semantically similar the keywords of each topic are to the corresponding representative keyword. To achieve this, the evaluators are asked to study the generated fine-grained topics in each domain and to label their keywords as "relevant" or "irrelevant". Table 8.12 shows an example of this evaluation.

Table 8.12: Example of the evaluated fine-grained topic in terms of the semantic consistency to the representative keyword

Representative keyword	
isolation toiture	
isolant thermique	relevant
isolation maison	relevant
isolation du sol	irrelevant
guide isolation	relevant
isolation plafond	irrelevant

We note that only topics with good representative keywords are passed to the evaluators. Using the evaluation labels, we firstly compute the precision on each topic separately and then average them to get the precision value over the 10 domains, as shown in Table 8.13. The average precision value over all the domains is 91.3%.

Table 8.13: Evaluation of the semantic consistency of the fine-grained topics

Domain	*Precision*
Abri de jardin	0.79
Isolation	0.95
Robinetterie	0.85
Assurance automobile	0.83
Certification	0.77
Collier de serrage	0.81
Pâtes	0.77
Peinture murale	0.83
Protection auditive	0.78
Tableaux électriques	0.92
AVERAGE	0.913

8.3 Comparing with a baseline approach

In this section, we compare our topic detection approach with respect to the state of the art. We choose KeyGraph (Sayyadi and Raschid, 2013) as the baseline approach, which is a graph-based approach for detecting the underlying topics of a collection of keywords.

Although Latent Dirichlet allocation (LDA) (Blei et al., 2003) has been widely used for this purpose, we do not aim to compare our approach with this topic modeling approach. As discussed in Chapter 3, we found the topics returned by KeyGraph to be more semantically consistent than LDA and the experiments performed by (Sayyadi and Raschid, 2013) verify this finding. Unlike the original model of LDA, which is word-level and does not support multi-token lexical units, KeyGraph supports both types of units. Since our topic detection approach is applied on a collection of single and multi-token keywords and the goal is to detect semantically consistent topics, we found KeyGraph to be a better baseline approach for our experiments.

KeyGraph is available open source under *GPLv2* license[2]. It accepts two types of input: a set of documents or a set of documents along with the associated keywords and their frequencies. In the first case, KeyGraph's keyword extraction is performed on the input documents, which extracts words, noun phrases, and named entities. In the second case, however, keywords are initially extracted using any desired extraction approach and only the topic detection step of KeyGraph is executed. We note that in KeyGraph, once detected, the topics are assigned to the input documents. Since this step is out of scope of this thesis, we do not take it into consideration in our experiment. Here, we target the second type of input and merely focus on the topic detection step of KeyGraph. This is to make the comparison on the keywords extracted using our keyword extraction approach. However, in the future, it would be interesting to evaluate our topic detection approach on the keywords extracted by another approach and from other types of documents.

The output of KeyGraph is a set of topics. Each topic is characterized by a list of keywords, used in our evaluation, and a set of documents that represent the topic. The detected topics can overlap.

To compare KeyGraph with our approach, we adapted the way it captures the occurrence of a keyword in a document. In KeyGraph, the keywords extracted from a document are either single words or sequences of words which are adjacent in the document. Hence, their occurrences can be found through a simple matching between the keywords and the documents. However, our keyword extraction approach may generate keywords in which tokens are not necessarily adjacent in documents. Consequently, a simple match cannot capture their occurrences in the documents. To adapt KeyGraph to this property of our approach, we rather use a window-wise matching. According to this matching, a keyword occurs in a document if all pairs of its adjacent tokens appear within a window of a fixed size in the document. As in Section 5.3, two tokens are assumed to be in the same window if there are at most two non-stop words between them. Using this rule, the frequency of keywords and their co-occurrences in the collection of documents is obtained. We recall that unlike our approach, KeyGraph generates a weighted graph, where the number of co-occurrences between two keywords represents the weight of their corresponding edge.

There are several parameters in KeyGraph that need to be fixed according to the target application and the type of documents. Authors have already recommended effective

[2]https://keygraph.codeplex.com/

parameters values for news, blogs, and tweets. Since in our enrichment approach, different types of web pages could be analyzed, from informative pages to commercial ones, finding effective values which suit all different types is not trivial. In this thesis, we do not study the best values of these parameters for our problem. As an alternative, we individually ran KeyGraph with each of the recommended parameters values on a sample input and found the blog parameters to perform the best for our problem. Therefore, we use the blog parameters for the experiments presented in this section.

Keywords in the topics detected by KeyGraph are not ranked and the topics are not labeled. Due to these properties, we cannot perform the same series of experiments that we did for evaluating our approach, *i.e.* evaluating representative keywords and evaluating the semantic consistency of a topic keywords with respect to its representative keyword. We point out that the level of granularity is also different in the topics detected by the two approaches. As a consequence, we only evaluate the effectiveness of KeyGraph in detecting out of topic keywords. This functionality is highly important in the enrichment application, as recommending keywords from other domains than the target one adds wrong information to the input document, which needs to be enriched.

More specifically, Figure 8.3 shows that for each domain of study, we pass a list of keywords and a collection of corresponding documents to KeyGraph. The keywords are the same as the ones passed to our approach for evaluating its performance. In the set of the topics detected by KeyGraph, we select the ones that are related to the domain and accordingly we get the list of the out of topic keywords. Comparing these keywords with the ones in the gold standard set, generated in 8.2.1, we measure the effectiveness of KeyGraph in detecting out of topic keywords.

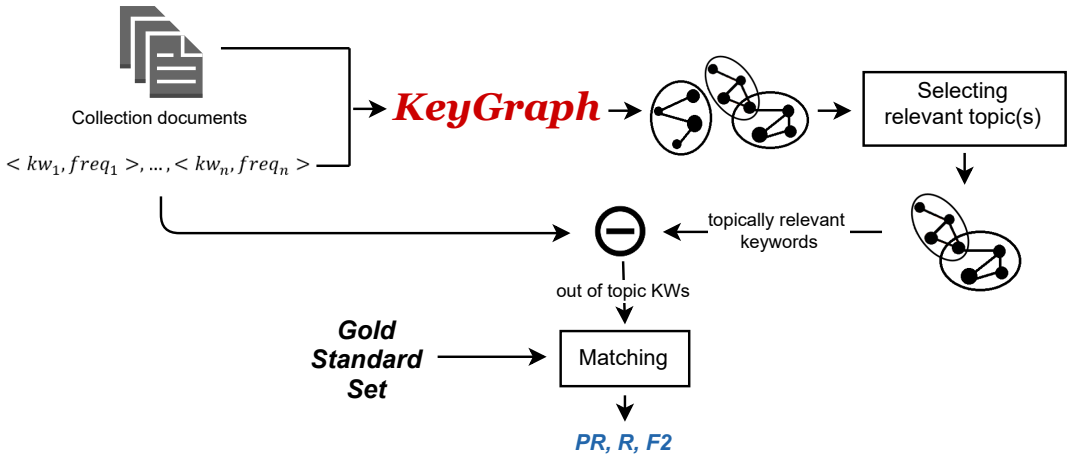


Figure 8.3: Protocol of evaluation for KeyGraph

Table 8.14 shows the values of precision, recall and F_2 measures obtained over the 10 domains. We recall that keywords of each domain have been extracted from 20 documents. We initially considered these documents as the input of KeyGraph. Intuitively, the size of this collection is not enough for capturing the co-occurrences between the input keywords.

Hence, we performed the experiments with two scenarios: using the 20 documents as the input collection and increasing the number of the collection documents to 500 to see how it affects the performance of KeyGraph. To collect more documents, we performed the same procedure as in 4.2.1 but selected more documents in the search engine result page. Table 8.14 reports the evaluation measures on both scenarios. According to the results, by increasing the number of the documents, generally, the performance slightly increases. However, considering the trade-off between the complexity of analysis and the performance of the approach, we did not find this increase to be effective.

Table 8.14: Evaluation of the out of topic keywords detected by KeyGraph

Domain	20 documents			500 documents		
	Precision	*Recall*	F_2	*Precision*	*Recall*	F_2
Abri de jardin	0.71	0.63	0.64	0.74	0.65	0.66
Isolation	0.49	0.76	0.68	0.49	0.8	0.71
Robinetterie	0.32	0.71	0.57	0.3	0.74	0.57
Assurance automobile	0.69	0.45	0.48	0.73	0.52	0.55
Certification	0.57	0.66	0.63	0.55	0.64	0.61
Collier de serrage	0.58	0.73	0.69	0.64	0.75	0.72
Pâtes	0.61	0.54	0.55	0.59	0.51	0.52
Peinture murale	0.53	0.75	0.69	0.48	0.76	0.68
Protection auditive	0.42	0.67	0.59	0.46	0.7	0.63
Tableaux électriques	0.32	0.59	0.5	0.39	0.61	0.54
AVERAGE	0.524	0.649	0.602	0.537	0.668	0.619

The change in the performance of KeyGraph after increasing the number of collection documents depends on the content of these documents. As Table 8.14 shows, in most of the domains, having more documents increases the co-occurrence within the same document of semantically similar keywords. This lets the co-occurrence feature better capture their similarity and consequently increases the recall value. On the contrary, we observed that for two domains "Certification" and "Pâtes", by adding more content, out of topic keywords are stronger connected to the relevant ones and fewer number of out of topic keywords are detected, which lowers the recall value. In addition, we observed that "Robinetterie" and "Peinture murale" domains achieve better recall values in exchange for precision. This shows that in these domains, by adding more content, the connectivity between a subset of semantically similar keywords is becoming stronger and as a result, other relevant keywords with weaker connectivities are detected as out of topic keywords. This indicates a loss of enrichment information. In most of the domains, however, the precision value increases by adding more content. This is due to the fact that similar keywords appear more often in the documents and so their connectivities are getting stronger.

It should be mentioned that since not all the pages returned by Google are necessarily related to the domain of study, the collection of 500 documents may not be rich enough for capturing the co-occurrences. By choosing another collection of documents, the effectiveness could increase more. Nevertheless, we did not find such collections available for our target domains.

In Figure 8.4, we compare our proposed approach with KeyGraph in terms of all the evaluation measures. For this comparison, we pick the better performing scenario in KeyGraph, *i.e.* with 500 collection documents. Our approach significantly outperforms KeyGraph specially in terms of recall value. This is justified by the fact that unlike KeyGraph, we do not exploit the explicit co-occurrences of keywords within documents and instead we focus on their semantic similarity. As a result, if two keywords co-occur frequently in a collection of documents but do not share any semantics, we do not detect them as relevant keywords.

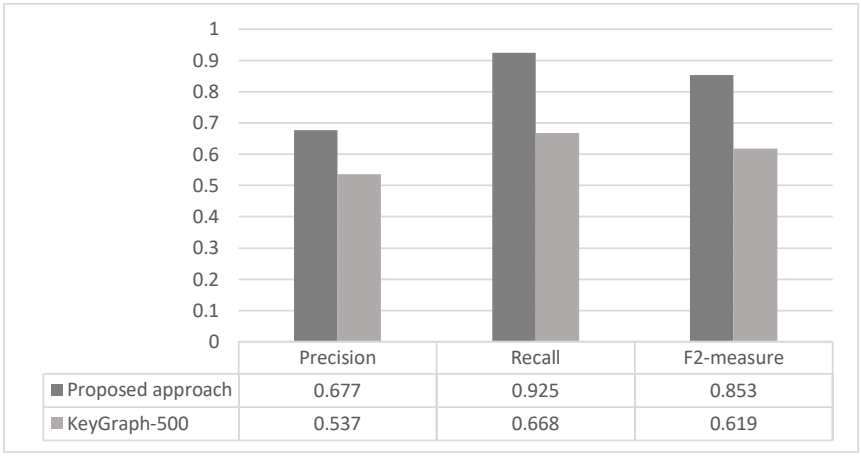


Figure 8.4: Effectiveness of the proposed approach *vs.* KeyGraph over 10 domains. In KeyGraph, the size of the collection for capturing the co-occurrence between keywords is 500.

The complexity of KeyGraph depends on the number of the input keywords and the size of the input collection. In our experiments, the average number of the input keywords is 185 and the maximum size of the collection is 500. With this size of data, we observed that the complexity of KeyGraph is lower than our approach, as it only exploits the collection of documents but we expand the context of each keyword to better capture the similarity between them. Since our topic detection approach is nevertheless executed in a reasonable time, due to its higher performance, we found it more effective for our enrichment application, where the semantic consistency of the recommended keywords must be assured.

8.4 Conclusion

In this chapter, we evaluated our proposed topic detection approach. We firstly evaluated the effectiveness of the similarity measures proposed in this thesis. Since generating a gold standard set of topics is very challenging, we do not explicitly evaluate the quality of the detected coarse-grained and fine-grained topics and rather we focus on three outputs of the approach, which we found highly important for our recommendation application.

Experiments on the similarity measures show that comparing to page contents, snippets are a better source of information for capturing the semantic similarity in the vocabulary-based similarity measure. In addition, we found the frequency to be a more effective feature for weighting the words in the vocabularies associated to two keywords. This feature is also robust across different domains. The results also showed that the co-occurrence-based similarity performs better when page contents are exploited as a source of information. Moreover, taking an average over the asymmetric similarities of co-occurrence-based measure is an effective way to make them symmetric. We showed that exploiting both vocabulary-based and co-occurrence-based similarities outperforms using each individually.

Our experiments on the coarse-grained topic detection approach shows that it can effectively detect the keywords which are not related to the domain of study ($F_2 = 85.3\%$). We also showed that our topic detection approach can effectively label the detected fine-grained topics with representative and dissimilar keywords and consequently can return discriminative topics ($F_{0.5} = 70.5\%$). This functionality of the approach can be however improved. Our analysis on the "bad" representative keywords in all the domains shows that 52% of them are representative of the domains but they are not discriminative enough with respect to the other representative keywords. This shows that the evaluation measures could be enhanced by improving the DFT algorithm as a future work.

Furthermore, we observed that the semantic consistency of the detected fine-grained topics with "good" representative keywords is highly assured with the average precision of 91.3%. This property of the approach is very important for our enrichment application. The high precision also indicates that the detected topics are mostly labeled properly, since there is a semantic consistency between the topic keywords and its representative keyword.

We compared the effectiveness of our approach with respect to KeyGraph, taken as baseline approach. Using a collection of web pages returned by Google as source of information in KeyGraph, our approach considerably outperformed KeyGraph in terms of precision, recall and F_2 measures. This shows that the explicit co-occurrences between keywords do not reflect their semantic similarity. As a result, KeyGraph may detect two dissimilar keywords as similar ones, since they are expected to co-occur in the same documents.

Our topic detection approach is domain-independent and our results show its effectiveness across different domains. Nevertheless, we observed that the performance of the approach could change from one domain to another one. This is due to the fact that the contexts provided for different domains are not equally rich. Clearly, the richer the context is, the more effective the approach will be. As a future work, it would be interesting to study the properties of the domains in order to tune the approach accordingly and to increase its performance.

Part VI

General conclusion

Chapter 9

Conclusion and perspectives

Conclusion

As mentioned in the introduction of this thesis, there are two types of semantic gap between an input document and a source of information: a gap in vocabulary and an informational gap. Depending on the target application, the source of information can be a term (query), a collection of documents, or a domain of interest. Both types of gap impact the accessibility and understanding of documents by users. Hence, it is a necessity to extend the vocabulary of the document or to add relevant information to it in order to fill in the semantic gap. However, since the gap may be large, due to the constraint on the length of the input document, the semantic gap is rather minimized by adding the most critical vocabulary and pieces of information, which are highly used in or covered by the source of information.

In this thesis, we proposed a domain-independent approach of enrichment in order to minimize the semantic gap between an input unstructured document and a target domain. More specifically, we answered the following research questions: how the important information of a domain can be retrieved and how its different topics can be detected so that only the one(s) related to the target domain can be processed afterwards? In other words, we showed how the most common vocabulary and the most critical information of the target domain can be detected, disambiguated and added to the input document. We focused on enriching web pages as a type of unstructured documents. The enrichment amounts to recommending metadata to a user, who aims at enriching a web page with respect to a domain.

In the initial chapters, we presented the definition of the problem and the corresponding state of the art approaches. Towards the middle of the thesis, we presented our enrichment approach. It starts with generating an enrichment collection, which consists of the important documents related to a domain of study. We divided the problem into two main components, keyword extraction and topic detection, which respectively extracts the information of the documents in the enrichment collection and identifies the information related to the target domain. These approaches have been tested in a real application context and on a wide range of various domains. In this thesis, we evaluated them through user-based evaluations on real applications and also by comparing with baseline approaches.

We proposed a keyword extraction method, which is adapted to unstructured and small textual data. Evaluation of the keyword extraction approach showed that in spite of its basic methodology, this approach is robust to various domains and also to the noise in web

pages. We also showed a significant improvement comparing to a modified version of the TF.IDF approach. Due to its shallow analysis, our approach can extract keywords of a web page in a very reasonable time.

The main contribution in the topic detection approach was to propose new similarity measures for capturing the semantic similarity and the semantic relatedness between two keywords. Our proposed similarities make use of the context returned by search engines as a rich source of information but they do not rely on the functionalities of search engines, which could change over time. Our evaluations confirmed the effectiveness of our proposed similarities in terms of both the exploited source of information and the chosen measure or function. We also showed that unlike the works in the state of the art, which either focus on the semantic similarity or the semantic relatedness between keywords, we take both into consideration and consequently better detect their relation.

Evaluations on the topic detection approach showed that it can effectively identify the keywords which do not belong to the studied domain. It also achieved a higher performance than the KeyGraph approach. Our approach therefore results in recommending relevant information to users and minimizing the semantic gap between the input document and the target domain. We also showed that the majority of the topics returned by our approach are discriminative: each one covers a unique piece of information within the domain of interest.

Comparing to the approaches in the state of the art, we are able to detect semantically consistent topics, which also have a good level of granularity so that users can interpret them easily. Keywords within each topic are ranked based on their importance within the topic. This ranking helps users to control the length of the input document in the enrichment procedure.

Since the keyword extraction and topic detection approaches can be easily adapted to different languages, we have implemented and tested them on more than one language even if only experiments on French are reported here. The main complexity of our proposed enrichment approach corresponds to the topic detection approach and, more specifically, to the similarity computation step. Nevertheless, although this approach is not interacting with users in real time, it is executed in a reasonable time and can be used as a recommendation application.

We believe that the documents enriched using our proposed enrichment approach could be further used in different text analysis tasks, such as classification and search applications, in order to provide more contextual information for NLP tools and to consequently improve their performance.

Perspectives

In the future, the following ideas could be addressed:

The performance of the proposed keyword extraction approach could be enhanced by improving the extraction features. More specifically, we could exploit new extraction features, which do not rely on frequencies, in order to capture the importance of rare but important words in documents and to increase the recall of the approach. Additional ill-formed patterns could also be identified in order to improve the post-filtering step.

In the short future, we could try to use the third order similarity measures in the vocabulary-based similarity in order to see how the accuracy of the similarity would change by considering the collocation between words rather than their exact matches in the snippets returned by Google.

Nowadays, word embedding models and deep learning algorithms are widely used to find the similarity between words or pieces of text. As a future work, we are interested to see how these models and algorithms could improve the similarity calculation: Do they enhance the overall similarity if we use them as the third similarity measure? Do they mainly focus on the semantic similarity or the semantic relatedness? Can we find or generate a rich and representative training data that can be used effectively for any domain? Knowing that it takes a long time to train a classifier in a deep learning model, does it keep the execution time of our enrichment approach reasonable? Our preliminarily research on the text similarity using word embedding models did not return remarkable results and, surprisingly, we found TF.IDF to perform better than word2vec (Mikolov et al., 2013). Studies such as that of Sahlgren and Lenci (2016) discuss that neural network-based models are not effective if the size of the training data is not big enough. Hence, as a future work, we could try a bigger and a more representative training data to study the performance of neural network-based models for our application.

Our evaluation on the representative keywords returned by the topic detection approach showed that roughly half of the "bad" representative keywords are in fact representative but not discriminative enough. We believe that this is mostly related to the similarity threshold that we determined empirically in the Discriminative Fine-grained Topics (DFT) algorithm for considering two keywords to be dissimilar. Although this threshold value is effective in many domains, in some others, another value could better detect the dissimilarity between representative keywords. Our intuition is that different properties of a domain, such as the amount of available context retrieved from Google, the level of connectivity between keywords in the graph-based model, the genericity or specificity of the domain, etc., could impact the optimal value of the similarity threshold. Hence, by studying these properties, we could tune the threshold value accordingly so that to return more discriminative topics.

Tuning threshold values depending on domain properties could be also tried for the similarity thresholds. In our preliminary research, we tried to generate a generic training data in order to learn the weights of the similarity measures and to automatically predict the similarities in different domains. On the contrary, in the future, our goal would be to automatically generate domain-specific training sets based on properties of each domain and to find the optimal values accordingly. The whole approach would remain domain-independent as domain tuning would be automatic.

We believe that the topic detection approach can be applied on any other type of documents conditioning that the input keywords are available *a priori*. This claim, however, has not been justified through formal experiments. As a future work, this property should be studied in order to confirm its applicability on other applications.

In the future, we can expand our recommendation application by generating pieces of text out of the recommended metadata. For this purpose, Natural Language Generation (NLG) techniques must be exploited. These techniques can be template-based, in which the input metadata fits into existing templates. The techniques can be also more advanced by interpreting the data and dynamically creating text. Our intuition is that generating templates for different domains and various types of web pages is not a trivial task and alternatively, advanced NLG techniques need to be exploited for our enrichment application.

Bibliography

Willyan D. Abilhoa and Leandro N. de Castro. A keyword extraction method from twitter messages represented as graphs. *Applied Mathematics and Computation*, 240:308–325, 2014. ISSN 0096-3003. URL http://search.ebscohost.com.gate6.inist.fr/login.aspx?direct=true&db=edselp&AN=S0096300314006304&lang=fr&site=eds-live.

Nitish Aggarwal, Kartik Asooja, and Paul Buitelaar. Deri&upm: Pushing corpus based relatedness to similarity: Shared task system description. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 643–647, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2387636.2387745.

Yong-Yeol Ahn, James P. Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010. doi: 10.1038/nature09182. URL http://dx.doi.org/10.1038/nature09182.

James Allan, editor. *Topic Detection and Tracking: Event-based Information Organization*. Kluwer Academic Publishers, Norwell, MA, USA, 2002. ISBN 0-7923-7664-1.

James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 37–45, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. doi: 10.1145/290941.290954. URL http://doi.acm.org/10.1145/290941.290954.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. Semeval 2017 task 10: Scienceie - extracting keyphrases and relations from scientific publications. *CoRR*, abs/1704.02853, 2017. URL http://arxiv.org/abs/1704.02853.

Kayhan Batmanghelich, Ardavan Saeedi, Karthik Narasimhan, and Samuel Gershman. Nonparametric spherical topic modeling with word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*, 2016. URL http://aclweb.org/anthology/P/P16/P16-2087.pdf.

Eda Baykan, Monika Henzinger, Ludmila Marian, and Ingmar Weber. Purely url-based topic classification. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 1109–1110, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-487-4. doi: 10.1145/1526709.1526880. URL http://doi.acm.org/10.1145/1526709.1526880.

G. Bekoulis and F. Rousseau. Graph-based term weighting scheme for topic modeling. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 1039–1044, Dec 2016. doi: 10.1109/ICDMW.2016.0150.

István Bíró, Jácint Szabó, and András A. Benczúr. Latent dirichlet allocation in web spam filtering. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb '08, pages 29–32, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-159-0. doi: 10.1145/1451983.1451991. URL http://doi.acm.org/10.1145/1451983.1451991.

D. Blei and J. Lafferty. A correlated topic model of science. *Annals of Applied Statistics*, 1:17–35., 2007.

David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 113–120, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143859. URL http://doi.acm.org/10.1145/1143844.1143859.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=944919.944937.

D.M. Blei, T.L. Griffiths, M.I. Jordan, and J.B. Tenenbaum. Hierarchical Topic Models and the Nested Chinese Restaurant Process. *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*, 2004.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Measuring semantic similarity between words using web search engines. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 757–766, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: http://doi.acm.org/10.1145/1242572.1242675.

Florian Boudin. A comparison of centrality measures for graph-based keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 834–838, Nagoya, Japan, October 2013. URL http://www.aclweb.org/anthology/I13-1102.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction. In *Sixth International Joint Conference on Natural Language Processing (IJCNLP)*, pages 543–551, 2013. URL http://aclweb.org/anthology/I/I13/I13-1062.pdf.

Didier Bourigault, Christian Jacquemin, and Marie-Claude L'Homme, editors. *Recent Advances in Computational Terminology*. John Benjamins, 2001.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, April 1998. ISSN 0169-7552. doi: 10.1016/S0169-7552(98)00110-X. URL http://dx.doi.org/10.1016/S0169-7552(98)00110-X.

Davide Buscaldi, Ronan Tournier, Nathalie Aussenac-Gilles, and Josiane Mothe. Irit: Textual similarity combining conceptual similarity with an n-gram comparison method. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 552–556, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2387636.2387729.

Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. Citation-enhanced keyphrase extraction from research papers: A supervised approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1435–1446, 2014. URL http://aclweb.org/anthology/D/D14/D14-1150.pdf.

Claudio Carpineto and Giovanni Romano. Optimal meta search results clustering. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 170–177, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0153-4. doi: 10.1145/1835449.1835480. URL http://doi.acm.org/10.1145/1835449.1835480.

C. Carretero-Campos, P. Bernaola-Galvan, A.V. Coronado, and P. Carpena. Improving statistical keyword detection in short texts: entropic and clustering approaches. *Physica A: Statistical Mechanics and its Applications*, 392(6):1481–1492, 2013. URL http://search.ebscohost.com.gate6.inist.fr/login.aspx?direct=true&db=inh&AN=13772872&lang=fr&site=eds-live.

Peggy Cellier, Thierry Charnois, Andreas Hotho, Stan Matwin, Marie-Francine Moens, and Yannick Toussaint, editors. *Proceedings of the 1st Int. Workshop on Interactions between Data Mining and Natural Language Processing (DMNLP@PKDD/ECML)*, volume 1202 of *CEUR Workshop Proceedings*, Nancy, France, 2014. URL http://ceur-ws.org/Vol-1202.

Ru-Yng Chang and Chung-Hsien Wu. Propositional term extraction over short text using word cohesiveness and conditional random fields with multi-level features. In *Proceedings of the 20th Conference on Computational Linguistics and Speech Processing (ROCLING)*, Taipei, Taiwan, 2008. ACLCLP. URL http://dblp.uni-trier.de/db/conf/rocling/rocling2008.html#ChangW08.

Dingyi Chen, Xue Li, Jing Liu, and Xia Chen. Ranking-constrained keyword sequence extraction from web documents. In Athman Bouguettaya and Xuemin Lin, editors, *Proceedings of the 20th Australasian Database Conference (ADC)*, volume 92, pages 161–169, Wellington, New Zealand, 2009. ACS.

Hsin-Hsi Chen, Ming-Shun Lin, and Yu-Chuan Wei. Novel association measures using web search with double checking. In *Proceedings of the 21st International Con-*

ference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, ACL-44, pages 1009–1016, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220302. URL http://dx.doi.org/10.3115/1220175.1220302.

Rudi L. Cilibrasi and Paul M. B. Vitanyi. The google similarity distance. *IEEE Trans. on Knowl. and Data Eng.*, 19(3):370–383, March 2007. ISSN 1041-4347. doi: 10.1109/TKDE.2007.48. URL http://dx.doi.org/10.1109/TKDE.2007.48.

Philipp Cimiano, Siegfried Handschuh, and Steffen Staab. Towards the self-annotating web. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 462–471, New York, NY, USA, 2004. ACM. ISBN 1-58113-844-X. doi: 10.1145/988672.988735. URL http://doi.acm.org/10.1145/988672.988735.

Jonathan D. Cohen. Highlights: Language- and domain-independent automatic indexing terms for abstracting. *Journal of the American Society for Information Science (JASIS)*, 46(3):162–174, 1995.

L. M. Collins and C. W. Dent. Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions. *Multivariate Behavioral Research*, 23(2):231–242, 1988.

Courtney Corley and Rada Mihalcea. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, EMSEE '05, pages 13–18, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1631862.1631865.

Béatrice Daille. Study and implementation of combined techniques for automatic extraction of terminology. In Philip Resnik and Judith L. Klavans, editors, *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, pages 49–66. MIT Press, Cambridge, MA, USA, 1996.

Soheil Danesh, Tamara Sumner, and James H. Martin. SGRank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction. In Martha Palmer, Gemma Boleda, and Paolo Rosso, editors, *Proceedings of the 4th Joint Conference on Lexical and Computational Semantics (SEM)*, pages 117–126, Denver, Colorado, USA, 2015. ISBN 978-1-941643-39-6.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian lda for topic models with word embeddings. In *ACL (1)*, pages 795–804. The Association for Computer Linguistics, 2015. ISBN 978-1-941643-72-3. URL http://dblp.uni-trier.de/db/conf/acl/acl2015-1.html#DasZD15.

Kushal S. Dave and Vasudeva Varma. Pattern based keyword extraction for contextual advertising. In *Proc. of the 19th ACM Int. Conf. on Information and Knowledge Management (CIKM)*, pages 1885–1888, New York, USA, 2010. ISBN 978-1-4503-0099-5.

doi: 10.1145/1871437.1871754. URL http://doi.acm.org/10.1145/1871437.1871754.

Herve Déjean, Eric Gaussier, and Fatia Sadat. Bilingual terminology extraction: an approach based on a multilingual thesaurus applicable to comparable corpora. In *Proc. of the 19th Int. Conf. on Computational Linguistics (COLING'02)*, Taipei, Taiwan, 2002.

Gaël Dias, Elsa Alves, and José Gabriel Pereira Lopes. Topic segmentation algorithms for text summarization and passage retrieval: An exhaustive evaluation. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 1334–1339, 2007. URL http://www.aaai.org/Library/AAAI/2007/aaai07-211.php.

Zhuanlian Ding, Xingyi Zhang, Dengdi Sun, and Bin Luo. Overlapping community detection based on network decomposition. *Sci Rep*, 6:24115, Apr 2016. ISSN 2045-2322. doi: 10.1038/srep24115. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4828636/.

Martin Dostàl and Karel Jezek. Automatic keyphrase extraction based on nlp and statistical methods. In Václav Snásel, Jaroslav Pokorný, and Karel Richta, editors, *Proc. of the Annual Int. Workshop on DAtabases, TExts, Specifications and Objects (DATESO)*, volume 706 of *CEUR Workshop Proceedings*, pages 140–145, 2011. URL http://dblp.uni-trier.de/db/conf/dateso/dateso2011.html#DostalJ11.

Gabriel Doyle and Charles Elkan. Accounting for burstiness in topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 281–288, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553410. URL http://doi.acm.org/10.1145/1553374.1553410.

Mark Dredze, Hanna M. Wallach, Danny Puller, and Fernando Pereira. Generating summary keywords for emails using topics. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, IUI '08, pages 199–206, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-987-6. doi: 10.1145/1378773.1378800. URL http://doi.acm.org/10.1145/1378773.1378800.

Patrick Drouin. Term extraction using non-technical corpora as a point of leverage. *Terminology*, 9(1):99–115, 2003.

Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. Web question answering: Is more always better? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 291–298, New York, NY, USA, 2002. ACM. ISBN 1-58113-561-0. doi: 10.1145/564376.564428. URL http://doi.acm.org/10.1145/564376.564428.

David Eppstein, Maarten Löffler, and Darren Strash. *Listing All Maximal Cliques in Sparse Graphs in Near-Optimal Time*, pages 403–414. Springer Berlin Heidelberg, Berlin,

Heidelberg, 2010. ISBN 978-3-642-17517-6. doi: 10.1007/978-3-642-17517-6_36. URL http://dx.doi.org/10.1007/978-3-642-17517-6_36.

T. S. Evans and R. Lambiotte. Line graphs, link partitions and overlapping communities, 2009. URL http://arxiv.org/abs/0903.2181.

T. S. Evans and R. Lambiotte. Line graphs of weighted networks for overlapping communities. *The European Physical Journal B - Condensed Matter and Complex Systems*, 77 (2):265–272, September 2010. ISSN 1434-6028. doi: 10.1140/epjb/e2010-00261-8. URL http://dx.doi.org/10.1140/epjb/e2010-00261-8.

Yi Fang, Naveen Somasundaram, Luo Si, Jeongwoo Ko, and Aditya P. Mathur. Analysis of an expert search query log. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 1189–1190, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2010113. URL http://doi.acm.org/10.1145/2009916.2010113.

Illés Farkas, Dániel Ábel, Gergely Palla, and Tamás Vicsek. Weighted network modules. *New Journal of Physics*, 9(6):180, 2007.

J.L. Fleiss et al. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010. ISSN 0370-1573. doi: DOI:10.1016/j.physrep.2009.11.002. URL http://www.sciencedirect.com/science/article/B6TVP-4XPYXF1-1/2/99061fac6435db4343b2374d26e64ac1.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. Domain-specific keyphrase extraction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, IJCAI '99, pages 668–673, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-613-0. URL http://dl.acm.org/citation.cfm?id=646307.687591.

Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries*, 3 (2):115–130, August 2000. URL http://dx.doi.org/10.1007/s007999900023.

Sara Elena Garza Villarreal and Ramón F. Brena. Topic mining based on graph local clustering. In *Proceedings of the 10th International Conference on Artificial Intelligence: Advances in Soft Computing - Volume Part II*, MICAI'11, pages 201–212, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-25329-4. doi: 10.1007/978-3-642-25330-0_18. URL http://dx.doi.org/10.1007/978-3-642-25330-0_18.

Peter V. Gehler, Alex D. Holub, and Max Welling. The rate adapting poisson model for information retrieval and object recognition. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 337–344, New York, NY, USA, 2006.

ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143887. URL http://doi.acm.org/10.1145/1143844.1143887.

Ann Gledson and John Keane. Using web-search results to measure word-group similarity. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 281–288, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. ISBN 978-1-905593-44-6. URL http://dl.acm.org/citation.cfm?id=1599081.1599117.

G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970. ISSN 0945-3245. doi: 10.1007/BF02163027. URL http://dx.doi.org/10.1007/BF02163027.

Wael H. Gomaa and Aly A. Fahmy. Article: A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, April 2013.

Google. Google's search engine optimization starter guide. October 2010. URL https://webmasters.googleblog.com/2010/09/seo-starter-guide-updated.html.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.

Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. Extracting key terms from noisy and multitheme documents. In *Proc. of the 18th Int. Conf. on World Wide Web (WWW)*, pages 661–670, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-487-4. doi: 10.1145/1526709.1526798. URL http://doi.acm.org/10.1145/1526709.1526798.

Weiwei Guo, Hao Li, Heng Ji, and Mona T. Diab. Linking tweets to news: A framework to enrich short text data in social media. In *ACL (1)*, pages 239–249. The Association for Computer Linguistics, 2013. ISBN 978-1-937284-50-3. URL http://dblp.uni-trier.de/db/conf/acl/acl2013-1.html#GuoLJD13.

Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

Kazi Saidul Hasan and Vincent Ng. Automatic keyphrase extraction: A survey of the state of the art. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1262–1273, Baltimore, USA, June 2014. URL http://www.aclweb.org/anthology/P14-1119.

Yulan He. Extracting topical phrases from clinical documents. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2957–2963. AAAI Press, 2016. URL http://dl.acm.org/citation.cfm?id=3016100.3016316.

P. J. Herrera and A. P. Pury. Statistical keyword detection in literary corpora. *The European Physical Journal B*, 63(1):135–146, 2008. ISSN 1434-6036. doi: 10.1140/epjb/e2008-00206-x. URL http://dx.doi.org/10.1140/epjb/e2008-00206-x.

Donnald Hindle. Noun classification from predicate-argument structures. In *Proc. of the annual meeting of the Association for Computational Linguistics (ACL)*, pages 268–275, 1990.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Replicated softmax: an undirected topic model. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1607–1614. Curran Associates, Inc., 2009. URL http://papers.nips.cc/paper/3856-replicated-softmax-an-undirected-topic-model.pdf.

Graeme Hirst and David St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An electronic lexical database.*, pages 305–332, Cambridge, MA, 1998. MIT Press. ISBN 978-0262061971.

Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 289–296, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-614-9. URL http://dl.acm.org/citation.cfm?id=2073796.2073829.

Andreas Hotho, Steffen Staab, and Gerd Stumme. Wordnet improves text document clustering. In *In Proc. of the SIGIR 2003 Semantic Web Workshop*, pages 541–544, 2003.

Jian Hu, Lujun Fang, Yang Cao, Hua-Jun Zeng, Hua Li, Qiang Yang, and Zheng Chen. Enhancing text clustering by leveraging wikipedia semantics. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 179–186, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390367. URL http://doi.acm.org/10.1145/1390334.1390367.

Chong Huang, YongHong Tian, Zhi Zhou, Charles X. Ling, and Tiejun Huang. Keyphrase extraction using semantic networks structure analysis. In *Proc. of the 6th Int. Conf. on Data Mining (ICDM)*, pages 275–284. IEEE, 2006. ISBN 0-7695-2701-9. URL http://dblp.uni-trier.de/db/conf/icdm/icdm2006.html#HuangTZLH06.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2 (1):193–218, 1985. ISSN 1432-1343. doi: 10.1007/BF01908075. URL http://dx.doi.org/10.1007/BF01908075.

Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 216–223, Stroudsburg, PA, USA, 2003. ACL. doi: 10.3115/1119355.1119383. URL http://dx.doi.org/10.3115/1119355.1119383.

Richard Hussey, Shirley Williams, and Richard Mitchell. Keyphrase extraction by synonym analysis of n-grams for e-journals categorisation. In *Proc. of the 3rd Int. Conf. on Information, Process, and Knowledge Management (eKNOW)*, pages 83–86, 2011.

E. Iosif and A. Potamianos. Unsupervised semantic similarity computation between terms using web documents. *IEEE Transactions on Knowledge and Data Engineering*, 22(11): 1637–1647, Nov 2010. ISSN 1041-4347. doi: 10.1109/TKDE.2009.193.

A. Islam and D. Inkpen. Second order co-occurrence pmi for determining the semantic similarity of words. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1033–1038, 2006. URL http://scholar.google.de/scholar.bib?q=info:9L067B-TuY8J:scholar.google.com/&output=citation&hl=de&as_sdt=0,5&ct=citation&cd=0.

Aminul Islam and Diana Inkpen. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Trans. Knowl. Discov. Data*, 2(2):10:1–10:25, July 2008. ISSN 1556-4681. doi: 10.1145/1376815.1376819. URL http://doi.acm.org/10.1145/1376815.1376819.

C. Jacquemin and D. Bourigault. *Term Extraction and Automatic Indexing*. The Oxford Handbook of Computational Linguistics, Oxford University Press, 2003.

J.J. Jiang and D.W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, pages 19–33, 1997. URL http://www.cse.iitb.ac.in/~cs626-449/Papers/WordSimilarity/4.pdf.

Jaap Kamps, Maarten Marx, Robert J. Mokken, and Maarten de Rijke. Using WordNet to measure semantic orientation of adjectives. In *LREC 2004*, volume 4, pages 1115–1118, 2004. URL http://citeseer.ist.psu.edu/kamps04using.html.

Daniel Kelleher and Saturnino Luz. Automatic hypertext keyphrase detection. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proc. of the 19th Int. Joint Conf. on Artificial intelligence (IJCAI)*, pages 1608–1609, 2005. ISBN 0938075934. URL http://dblp.uni-trier.de/db/conf/ijcai/ijcai2005.html#KelleherL05.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 21–26, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1859664.1859668.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, September 1999. ISSN 0004-5411. doi: 10.1145/324133.324140. URL http://doi.acm.org/10.1145/324133.324140.

Peter Kolb. Experiments on the difference between semantic similarity and relatedness. In Kristiina Jokinen and Eckhard Bick, editors, *Proceedings of the 17th Nordic Conference*

of Computational Linguistics NODALIDA 2009, volume 4, pages 81–88. Northern European Association for Language Technology, 2009. URL http://dspace.utlib.ee/dspace/bitstream/10062/9731/1/paper37.pdf.

Saul Kripke. Naming and necessity. In G. Harman D. Davidson, editor, *Semantics of Natural Language*. Reidel, Dordrecht, 1972.

Giridhar Kumaran and James Allan. Text classification and named entities for new event detection. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 297–304, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4. doi: 10.1145/1008992.1009044. URL http://doi.acm.org/10.1145/1008992.1009044.

Jussi M. Kumpula, Mikko Kivelä, Kimmo Kaski, and Jari Saramäki. A sequential algorithm for fast clique percolation. *Physical Review E*, 78(2), July 2008. doi: 10.1103/physreve.78.026109. URL http://dx.doi.org/10.1103/physreve.78.026109.

Simon Lacoste-Julien, Fei Sha, and Michael I. Jordan. Disclda: Discriminative learning for dimensionality reduction and classification. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 897–904. Curran Associates, Inc., 2009.

Shibamouli Lahiri, Sagnik Ray Choudhury, and Cornelia Caragea. Keyword and keyphrase extraction using centrality measures on collocation networks. *CoRR*, abs/1401.6571, 2014. URL http://arxiv.org/abs/1401.6571.

Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3): 033015, 2009. URL http://stacks.iop.org/1367-2630/11/i=3/a=033015.

T.K. Landauer, P.W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse processes*, 25:259–284, 1998.

J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 1977.

Victor Lavrenko, James Allan, Edward DeGuzman, Daniel LaFlamme, Veera Pollard, and Stephen Thomas. Relevance models for topic detection and tracking. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 115–121, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1289189.1289268.

C. Leacock and M. Chodorow. Combining local context and wordnet similarity for word sense identification. In Christiane Fellfaum, editor, *MIT Press*, pages 265–283, Cambridge, Massachusetts, 1998.

Kathy Lee, Diana Palsetia, Ramanathan Narayanan, Md. Mostofa Ali Patwary, Ankit Agrawal, and Alok Choudhary. Twitter trending topic classification. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, ICDMW '11, pages 251–258, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4409-0. doi: 10.1109/ICDMW.2011.171. URL http://dx.doi.org/10.1109/ICDMW.2011.171.

Els Lefever, Lieve Macken, and Veronique Hoste. Language-independent bilingual terminology extraction from a multilingual parallel corpus. In *Proc. of the 12th Conf of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 496–504, Athens, Greece, 2009. ACL.

Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26, New York, NY, USA, 1986. ACM. ISBN 0-89791-224-1. doi: 10.1145/318723.318728. URL http://doi.acm.org/10.1145/318723.318728.

VI Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, 1966.

Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 577–584, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143917. URL http://doi.acm.org/10.1145/1143844.1143917.

Zhiwei Li, Bin Wang, Mingjing Li, and Wei-Ying Ma. A probabilistic model for retrospective news event detection. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 106–113, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5. doi: 10.1145/1076034.1076055. URL http://doi.acm.org/10.1145/1076034.1076055.

Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8. URL http://dl.acm.org/citation.cfm?id=645527.657297.

Dekang Lin and Shaojun Zhao. Identifying synonyms among distributionally similar words. In *Proceedings of IJCAI-03*, pages 1492–1493, 2003.

Robert V. Lindsey, William P. Headden, III, and Michael J. Stipicevic. A phrase-discovering topic model using hierarchical pitman-yor processes. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 214–222, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2390948.2390975.

Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proc. of Human Language Technologies (NAACL)*, pages 620–628, Boulder, Colorado, 2009. ACL. ISBN 978-1-932432-41-1. URL http://dl.acm.org/citation.cfm?id=1620754.1620845.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. Automatic keyphrase extraction via topic decomposition. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 366–376, Cambridge, Massachusetts, 2010. ACL. URL http://dl.acm.org/citation.cfm?id=1870658.1870694.

Rachel Tsz-Wai Lo, Ben He, and Iadh Ounis. Automatically building a stopword list for an information retrieval system. *JDIM*, 3(1):3–8, 2005. URL http://dblp.uni-trier.de/db/journals/jdim/jdim3.html#LoHO05.

H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM J. of Research and Devlopment*, 1(4):309–317, October 1957. ISSN 0018-8646. doi: 10.1147/rd.14.0309. URL http://dx.doi.org/10.1147/rd.14.0309.

Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208, 1996. ISSN 1532-5970. doi: 10.3758/BF03204766. URL http://dx.doi.org/10.3758/BF03204766.

Liang Ma, Tingting He, Fang Li, Zhuomin Gui, and Jinguang Chen. Query-focused multi-document summarization using keyword extraction. In *Proc. of the Int. Conf. on Computer Science and Software Engineering (CSSE)*, volume 1, pages 20–23, Washington, DC, USA, 2008. IEEE. ISBN 978-0-7695-3336-0. doi: 10.1109/CSSE.2008.1323. URL http://dx.doi.org/10.1109/CSSE.2008.1323.

Juha Makkonen, Helena Ahonen-Myka, and Marko Salmenkivi. Simple semantics in topic detection and tracking. *Inf. Retr.*, 7(3-4):347–368, September 2004. ISSN 1386-4564. doi: 10.1023/B:INRT.0000011210.12953.86. URL http://dx.doi.org/10.1023/B:INRT.0000011210.12953.86.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-13360-1.

Antonio Di Marco and Roberto Navigli. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39(3):709–754, 2013. URL http://dblp.uni-trier.de/db/journals/coling/coling39.html#MarcoN13.

Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13:2004, 2004.

Yutaka Matsuo, Yukio Ohsawa, and Mitsuru Ishizuka. Keyworld: Extracting keywords from a document as a small world. In *Proc. of the 4th Int. Conf. on Discovery Science (DS)*, volume 2226 of *LNCS*, pages 271–281, 2001. URL http://search.ebscohost.com.gate6.inist.fr/login.aspx?direct=true&db=fcs&AN=14046082&lang=fr&site=eds-live.

Jon D. Mcauliffe and David M. Blei. Supervised topic models. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 121–128. Curran Associates, Inc., 2008. URL http://papers.nips.cc/paper/3328-supervised-topic-models.pdf.

Olena Medelyan. *Human-competitive automatic topic indexing*. PhD thesis, The University of Waikato, 2009.

Olena Medelyan and Ian H. Witten. Thesaurus based automatic keyphrase indexing. In *Proc. of the 6th Joint Conference on Digital Libraries (JCDL)*, pages 296–297. ACM, 2006. ISBN 1-59593-354-9. doi: 10.1145/1141753.1141819. URL http://doi.acm.org/10.1145/1141753.1141819.

A. Mehri, M. Jamaati, and H. Mehri. Word ranking in a single document by jensen-shannon divergence. *Physics Letters A*, 379(28):1627–1632, 2015. URL http://search.ebscohost.com.gate6.inist.fr/login.aspx?direct=true&db=inh&AN=15395000&lang=fr&site=eds-live.

Ali Mehri and Amir H. Darooneh. The role of entropy in word ranking. *Physica A: Statistical Mechanics and its Applications*, 390:3157–3163, 2011. ISSN 0378-4371. URL http://search.ebscohost.com.gate6.inist.fr/login.aspx?direct=true&db=edselp&AN=S0378437111003074&lang=fr&site=eds-live.

Rada Mihalcea and Paul Tarau. TextRank: Bringing order into texts. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, July 2004.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 775–780. AAAI Press, 2006. ISBN 978-1-57735-281-5. URL http://dl.acm.org/citation.cfm?id=1597538.1597662.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013. URL http://arxiv.org/abs/1301.3781.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3: 235–244, 1990.

Nada Mimouni, Adeline Nazarenko, and Sylvie Salotti. Search and discovery in legal document networks. In *Legal Knowledge and Information Systems (28th JURIX conference)*, pages 187–188, 2015. doi: 10.3233/978-1-61499-609-5-187. URL http://dx.doi.org/10.3233/978-1-61499-609-5-187.

Saeedeh Momtazi, Sanjeev Khudanpur, and Dietrich Klakow. A comparative study of word co-occurrence for term clustering in language model-based sentence retrieval. In *Proc. of Human Language Technologies (NAACL)*, pages 325–328, Los Angeles, CA, USA, 2010. ACL. ISBN 1-932432-65-5. URL http://dl.acm.org/citation.cfm?id=1857999.1858045.

Jose G. Moreno. Text-based ephemeral clustering for web image retrieval on mobile devices. *SIGIR Forum*, 49(1):67–67, June 2015. ISSN 0163-5840. doi: 10.1145/2795403.2795419. URL http://doi.acm.org/10.1145/2795403.2795419.

Junichiro Mori, Yutaka Matsuo, Mitsuru Ishizuka, and Boi Faltings. Keyword extraction from the web for foaf metadata. In *Proc. of the Workshop on Friend of a Friend, Social Networking and the Semantic Web*, 2004.

Junichiro Mori, Takumi Tsujishita, Yutaka Matsuo, and Mitsuru Ishizuka. *Extracting Relations in Social Networks from the Web Using Similarity Between Collective Contexts*, pages 487–500. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-49055-5. doi: 10.1007/11926078_35. URL http://dx.doi.org/10.1007/11926078_35.

Alberto Muñoz. Compound key word generation from document databases using a hierarchical clustering {ART} model. *Intelligent Data Analysis*, 1:25–48, 1997. ISSN 1088-467X. doi: http://dx.doi.org/10.1016/S1088-467X(98)00008-0. URL http://www.sciencedirect.com/science/article/pii/S1088467X98000080.

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. URL http://www.ingentaconnect.com/content/jbp/li/2007/00000030/00000001/art00002. John Benjamins.

Roberto Navigli and Paola Velardi. Semantic interpretation of terminological strings. In *Proc. 6th Int'l Conf. on Terminology and Knowledge Engineering (TKE 2002)*, pages 95–100. Springer-Verlag, 2002.

Adeline Nazarenko and Touria Aït El Mekki. Building back-of-the-book indexes. In Fidelia Ibekwe-SanJuan, Anne Condamines, and M. Teresa Cabré Castellvi, editors, *Application-Driven Terminology Engineering*, pages 179–202. John Benjamins, 2007.

M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review*, E 69(026113), 2004.

Thuy Dung Nguyen and Min-Yen Kan. Keyphrase extraction in scientific publications. In *Proceedings of the 10th International Conference on Asian Digital Libraries: Looking Back 10 Years and Forging New Frontiers*, ICADL'07, pages 317–326, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3-540-77093-3, 978-3-540-77093-0. URL http://dl.acm.org/citation.cfm?id=1780653.1780707.

Yukio Ohsawa, Nels E. Benson, and Masahiko Yachida. KeyGraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In *Proc. of the Advances in Digital Libraries Conference (ADL)*, Washington, DC, USA, 1998a. IEEE. ISBN 0-8186-8464-X. URL http://dl.acm.org/citation.cfm?id=582987.785950.

Yukio Ohsawa, Nels E. Benson, and Masahiko Yachida. Keygraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In *ADL '98: Proceedings of the Advances in Digital Libraries Conference*, page 12, Washington, DC, USA, 1998b. IEEE Computer Society. ISBN 0-8186-8464-X. URL http://portal.acm.org/citation.cfm?id=785950.

Miguel Ortuño, Pedro Carpena, Pedro Bernaola-Galván, Enrique Muñoz, and Andrés M. Somoza. Keyword detection in natural languages and dna. *EPL (Europhysics Letters)*, 57 (5):759–764, 2002. URL http://stacks.iop.org/0295-5075/57/i=5/a=759.

Stanisław Osiński, Jerzy Stefanowski, and Dawid Weiss. *Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition*, pages 359–368. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-39985-8. doi: 10.1007/978-3-540-39985-8_37. URL https://doi.org/10.1007/978-3-540-39985-8_37.

J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan. Nested hierarchical dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):256–270, Feb 2015. ISSN 0162-8828. doi: 10.1109/TPAMI.2014.2318728.

Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435 (7043):814–818, June 2005. ISSN 0028-0836. URL http://dx.doi.org/10.1038/nature03607.

Gergely Palla, Illés J Farkas, Péter Pollner, Imre Derényi, and Tamás Vicsek. Directed network modules. *New Journal of Physics*, 9(6):186, 2007. URL http://stacks.iop.org/1367-2630/9/i=6/a=186.

Youngja Park, Roy J Byrd, and Branimir K Boguraev. Automatic glossary extraction: Beyond terminology identification. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1072228.1072370. URL https://doi.org/10.3115/1072228.1072370.

Siddharth Patwardhan. Incorporating Dictionary and Corpus Information into a Context Vector Measure of Semantic Relatedness. Master's thesis, University of Minnesota, August 2003. URL http://www.cs.utah.edu/~{}sidd/papers/Patwardhan03.pdf.

Siddharth Patwardhan. Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *In: Proceedings of the EACL*, pages 1–8, 2006.

J. Pearson. *Terms in Context*. Studies in corpus linguistics. J. Benjamins, 1998. ISBN 9789027222695. URL https://books.google.fr/books?id=OEOFd2lnNKkC.

S. Phuvipadawat and T. Murata. Breaking news detection and tracking in twitter. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 120–123, Aug 2010. doi: 10.1109/WI-IAT.2010.205.

Xiaoguang Qi, Lan Nie, and Brian D. Davison. Measuring similarity to detect qualified links. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb '07, pages 49–56, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-732-2. doi: 10.1145/1244408.1244418. URL http://doi.acm.org/10.1145/1244408.1244418.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1*, EMNLP '09, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-59-6. URL http://dl.acm.org/citation.cfm?id=1699510.1699543.

Joseph Reisinger, Austin Waters, Brian Silverthorn, and Raymond J. Mooney. Spherical topic models. In *ICML*, 2010. URL http://www.cs.utexas.edu/users/ml/papers/reisinger.icml10.pdf.

Ingrid Renz, Andrea Ficzay, and Holger Hitzler. Keyword extraction for text characterization. In Antje Düsterhöft and Bernhard Thalheim, editors, *Proc. of the 8th Int. Conf. on Applications of Natural Language to Information Systems*, volume 29 of *LNI*, pages 228–234, Burg (Spreewald), Germany, 2003. GI. ISBN 3-88579-358-X. URL http://dblp.uni-trier.de/db/conf/nldb/nldb2003.html#RenzFH03.

Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-363-8, 978-1-558-60363-9. URL http://dl.acm.org/citation.cfm?id=1625855.1625914.

R. Richardson, A. F. Smeaton, A. F. Smeaton, J. Murphy, and J. Murphy. Using wordnet as a knowledge base for measuring semantic similarity between words. Technical report, In Proceedings of AICS Conference, 1994.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. In Michael W. Berry and Jacob Kogan, editors, *Text Mining. Applications and Theory*, pages 1–20. John Wiley and Sons, Ltd, 2010. ISBN 9780470689646. doi: 10.1002/9780470689646.ch1. URL http://dx.doi.org/10.1002/9780470689646.ch1.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press. ISBN 0-9749039-0-6. URL http://dl.acm.org/citation.cfm?id=1036843.1036902.

J.C. Sager. *Practical Course in Terminology Processing*. John Benjamins Publishing Company, 1990. ISBN 9789027220769. URL https://books.google.com/books?id=Be4nBVIfj0wC.

Mehran Sahami and Timothy D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 377–386, New York, NY, USA, 2006. ACM. ISBN 1-59593-323-9. doi: 10.1145/1135777.1135834. URL http://doi.acm.org/10.1145/1135777.1135834.

Magnus Sahlgren and Alessandro Lenci. The effects of data size and frequency range on distributional semantic models. *CoRR*, abs/1609.08293, 2016. URL http://arxiv.org/abs/1609.08293.

G. Salton, C. S. Yang, and C. T. Yu. A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science*, 26(1):33–44, 1975.

Kamal Sarkar, Mita Nasipuri, and Suranjan Ghose. A new approach to keyphrase extraction using neural networks. *CoRR*, abs/1004.3274, 2010. URL http://dblp.uni-trier.de/db/journals/corr/corr1004.html#abs-1004-3274.

Hassan Sayyadi and Louiqa Raschid. A graph analytical approach for topic detection. *ACM Trans. Internet Technol.*, 13(2):4:1–4:23, December 2013. ISSN 1533-5399. doi: 10.1145/2542214.2542215. URL http://doi.acm.org/10.1145/2542214.2542215.

Hassan Sayyadi, Matthew Hurst, and Alexey Maykov. Event detection and tracking in social streams. In *In Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2009). AAAI*, 2009.

Ugo Scaiella, Paolo Ferragina, Andrea Marino, and Massimiliano Ciaramita. Topical clustering of search results. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 223–232, New York, NY, USA, 2012.

Satu Elisa Schaeffer. Survey: Graph clustering. *Comput. Sci. Rev.*, 1(1):27–64, August 2007. ISSN 1574-0137. doi: 10.1016/j.cosrev.2007.05.001. URL http://dx.doi.org/10.1016/j.cosrev.2007.05.001.

Markus Schedl, Tim Pohle, Peter Knees, and Gerhard Widmer. Bassigning and visualizing music genres by web-based co-occurrence analysis. In *in Proc. Int. Conf. Music Information Retrieval*, 2006.

SEOmoz. The beginners guide to SEO. Technical report, 2012.

Claude E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

Sushma S Nandgaonkar Sheetal A Takale. Measuring Semantic Similarity between Words Using Web Documents. *International Journal of Advanced Computer Science and Applications(IJACSA)*, 1(4), 2010. URL http://ijacsa.thesai.org/.

Ayush Singhal, Ravindra Kasturi, Ankit Sharma, and Jaideep Srivastava. Leveraging web resources for keyword assignment to short text documents. *CoRR*, abs/1706.05985, 2017. URL http://arxiv.org/abs/1706.05985.

Franck Smadja. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19 (1):143–177, march 1993. Special Issue on Using Large Corpora: I.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

Tadej Stajner, Delia Rusu, Lorand Dali, Blaz Fortuna, Dunja Mladenic, and Marko Grobelnik. A service oriented framework for natural language text enrichment. *Informatica (Slovenia)*, 34(3):307–313, 2010. URL http://dblp.uni-trier.de/db/journals/informaticaSI/informaticaSI34.html#StajnerRDFMG10.

Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. Creation and evaluation of large keyphrase extraction collections with multiple opinions. *Language Resources and Evaluation*, Jun 2017. ISSN 1574-0218. doi: 10.1007/s10579-017-9395-6. URL https://doi.org/10.1007/s10579-017-9395-6.

Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2004.

G. Tur and R. De Mori. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. Wiley, 2011. ISBN 9781119993940. URL https://books.google.com/books?id=RDLyT2FythgC.

Peter D. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2 (4):303–336, May 2000. ISSN 1386-4564. doi: 10.1023/A:1009976227802. URL http://dx.doi.org/10.1023/A:1009976227802.

Peter D. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, pages 491–502, London, UK, UK, 2001. Springer-Verlag. ISBN 3-540-42536-5. URL http://dl.acm.org/citation.cfm?id=645328.650004.

Peter D. Turney. Coherent keyphrase extraction via web mining. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 434–439. Morgan Kaufmann, 2003. URL http://dl.acm.org/citation.cfm?id=1630659.1630724.

Maisa Vidal, Guilherme V. Menezes, Klessius Berlt, Edleno S. de Moura, Karla Okada, Nivio Ziviani, David Fernandes, and Marco Cristo. Selecting keywords to represent web pages using wikipedia information. In *Proceedings of the 18th Brazilian Symposium on Multimedia and the Web*, WebMedia '12, pages 375–382, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1706-1. doi: 10.1145/2382636.2382714. URL http://doi.acm.org/10.1145/2382636.2382714.

Anthony J. Viera and Joanne M. Garrett. Understanding interobserver agreement: The kappa statistic. *Family Medicine*, 37(5):360–363, 5 2005. ISSN 0742-3225.

Hanna M. Wallach. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 977–984, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143967. URL http://doi.acm.org/10.1145/1143844.1143967.

Shen Wan and R. A. Angryk. Measuring semantic similarity using wordnet-based context vectors. In *2007 IEEE International Conference on Systems, Man and Cybernetics*, pages 908–913, Oct 2007. doi: 10.1109/ICSMC.2007.4413585.

Xiaojun Wan. Timedtextrank: Adding the temporal dimension to multi-document summarization. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 867–868, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. doi: 10.1145/1277741.1277949. URL http://doi.acm.org/10.1145/1277741.1277949.

Xiaojun Wan and Jianguo Xiao. Single document keyphrase extraction using neighborhood knowledge. In *Proc. of the 23rd National Con. on Artificial Intelligence (AAAI)*, pages 855–860, 2008a. ISBN 978-1-57735-368-3. URL http://dl.acm.org/citation.cfm?id=1620163.1620205.

Xiaojun Wan and Jianguo Xiao. CollabRank: Towards a collaborative approach to single-document keyphrase extraction. In *Proc. of the 22nd Int. Conf. on Computational Linguistics (COLING)*, pages 969–976, Manchester, UK, August 2008b. URL http://www.aclweb.org/anthology/C08-1122.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 552–559, Prague, Czech Republic, June 2007. ACL. URL http://www.aclweb.org/anthology/P07-1070.

Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pages 697–702, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-3018-4. doi: 10.1109/ICDM.2007.86. URL http://dx.doi.org/10.1109/ICDM.2007.86.

Christian Wartena and Rogier Brussee. Topic detection by clustering keywords. In *Proceedings of the 2008 19th International Conference on Database and Expert Systems Application*, DEXA '08, pages 54–58, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3299-8. doi: 10.1109/DEXA.2008.120. URL http://dx.doi.org/10.1109/DEXA.2008.120.

Xing Wei and W. Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 178–185, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: 10.1145/1148170.1148204. URL http://doi.acm.org/10.1145/1148170.1148204.

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. KEA: Practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, DL '99, pages 254–255, New York, NY, USA, 1999. ACM. ISBN 1-58113-145-3. doi: 10.1145/313238.313437. URL http://doi.acm.org/10.1145/313238.313437.

Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. doi: 10.3115/981732.981751. URL http://dx.doi.org/10.3115/981732.981751.

Jierui Xie, Stephen Kelley, and Boleslaw K. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Comput. Surv.*, 45 (4):43:1–43:35, August 2013. ISSN 0360-0300. doi: 10.1145/2501654.2501657. URL http://doi.acm.org/10.1145/2501654.2501657.

Pengtao Xie and Eric P. Xing. Integrating document clustering and topic modeling. *CoRR*, abs/1309.6874, 2013. URL http://dblp.uni-trier.de/db/journals/corr/corr1309.html#XieX13.

Eric P. Xing, Rong Yan, and Alexander G. Hauptmann. Mining associated text and images with dual-wing harmoniums. In *Proceedings of the Twenty-First Conference*

on Uncertainty in Artificial Intelligence, UAI'05, pages 633–641, Arlington, Virginia, United States, 2005. AUAI Press. ISBN 0-9749039-1-4. URL http://dl.acm.org/citation.cfm?id=3020336.3020413.

Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 28–36, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. doi: 10.1145/290941.290953. URL http://doi.acm.org/10.1145/290941.290953.

Wen-tau Yih, Joshua Goodman, and Vitor R. Carvalho. Finding advertising keywords on web pages. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 213–222, New York, NY, USA, 2006. ACM. ISBN 1-59593-323-9. doi: 10.1145/1135777.1135813. URL http://doi.acm.org/10.1145/1135777.1135813.

Z. Yu, T. R. Johnson, and R. Kavuluru. Phrase based topic modeling for semantic information processing in biomedicine. In *2013 12th International Conference on Machine Learning and Applications*, volume 1, pages 440–445, Dec 2013. doi: 10.1109/ICMLA.2013.89.

Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 46–54, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. doi: 10.1145/290941.290956. URL http://doi.acm.org/10.1145/290941.290956.

Haïfa Zargayouna and Adeline Nazarenko. Evaluation of textual knowledge acquisition tools: a challenging task. In *Proc. of the Int. Conf. on Language Resources and Evaluation (LREC)*. ELRA, may 2010.

Chen Zhang, Hao Wang, Liangliang Cao, Wei Wang, and Fanjiang Xu. A hybrid term-term relations analysis approach for topic detection. *Know.-Based Syst.*, 93(C):109–120, February 2016. ISSN 0950-7051. doi: 10.1016/j.knosys.2015.11.006. URL http://dx.doi.org/10.1016/j.knosys.2015.11.006.

Chengzhi Zhang, Huilin Wang, Yao Liu, Dan Wu, Yi Liao, and Bo. Wang. Automatic keyword extraction from documents using conditional random fields. *Computational Information Systems*, 2008.

Kuo Zhang, Hui Xu, Jie Tang, and Juanzi Li. Keyword extraction using support vector machine. In *Proc. of the 7th Int. Conf. on Advances in Web-Age Information Management (WAIM)*, pages 85–96. Springer Verlag, 2006. ISBN 3-540-35225-2, 978-3-540-35225-9. doi: 10.1007/11775300_8. URL http://dx.doi.org/10.1007/11775300_8.

Jun Zhu, Amr Ahmed, and Eric Xing. MedLDA: Maximum Margin Supervised Topic Models. *Journal of Machine Learning Research*, 1:1–48, 2010. URL http://www.cs.cmu.edu/~{}junzhu/MedLDAc/MedLDA_draft.pdf.

Xiaojin Zhu and R. Rosenfeld. Improving trigram language modeling with the world wide web. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, volume 1, pages 533–536, 2001. doi: 10.1109/ICASSP.2001.940885.