Università degli Studi di Udine
Dipartimento di Scienze matematiche, informatiche e fisiche
Dottorato di Ricerca in Informatica e Scienze Matematiche e Fisiche

Ph.D. Thesis

Multilingual Keyphrase Extraction and Advanced Localisation Strategies

Candidate:
Dante Degl'Innocenti

Supervisor:
Prof. Carlo Tasso

Cycle XXIX – April 2017

Author's address:
Dipartimento di Scienze Matematiche, Informatiche e Fisiche
Universita degli Studi di Udine
Via delle Science, 206
33100 Udine – Italia
+39 0432 558446
deglinnocenti.dante@spes.uniud.it

Institute Contacts:
Dipartimento di Scienze Matematiche, Informatiche e Fisiche
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine – Italia

To my family.

Abstract

Due to the exponential growth of user-generated Web content and ever-increasing access of emerging countries to the Web, the demands for quality localised Information Access tools has grown stronger and stronger. Providing a quality Information Access nowadays implies, however, involving Adaptive Personalisation, Semantic Web, and Artificial Intelligence techniques to filter non-relevant, offensive, inappropriate, and harmful content that traditional Information Retrieval techniques are not able to filter. To allow such systems to operate with accuracy, Information Extraction and Knowledge Representation technologies are required; while a lot of effort has been put into developing such tools for English content, relatively little effort has been put into localising them. Localisation, as a matter of fact, implies a great deal of effort and overcoming several non-trivial challenges. First and foremost, localised Information Extraction tools must cope with different languages, which is a challenge few research works have tackled due to the lack of linguistic resources and best practices that affect several non-English idioms. Adopting the right language, however, is not enough and localised Knowledge Representation should also be culture sensitive, i.e. aware of the many cultural factors that influence people's perception and behavior, which is a topic that has been mostly neglected up to now by the Artificial Intelligence research community. In this thesis we present a comprehensive discussion of localisation of Information Extraction and Knowledge Representation techniques, introducing multilingual Keyphrase Extraction and culture-sensitive Semantic Relatedness as case studies of multilingual and multicultural knowledge-intensive applications. The several experiments performed show that the proposed techniques, framework, and systems are effective, efficient, and provide a powerful tool that can be proficiently integrated into different applications to address localization and multiculturality issues.

Acknowledgments

I would first like to express my special appreciation and thanks to my Ph.D. Supervisor Prof. Carlo Tasso for his mentoring, patience and helpful advice. A special thanks to my fellow doctoral students Dario De Nart and Marco Pavan for their feedback, cooperation and of course friendship; without their precious support it would not be possible to conduct this research. My sincere thanks also goes to my fellow labmates Marco Basaldella and Muhammad Helmy for the useful comments, remarks and engagement. Finally, I express my warm thanks to all my fellow doctoral students Eddy Maddalena, Nicola Prezza, Luca Rizzi and all Professors and staff of the *Department of Mathematics, Computer Science and Physics* of the *University of Udine*. Last but not the least, I would like to thank my family for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them. Thank you.

Dante

Contents

List of Figures

List of Tables

1
Introduction

Over the last years the Web has become more and more inclusive, participative, and, above all, larger: more and more users gain access to the Web from all around the world, social platforms allow anyone to publish content, and popular services, such as search engines, are becoming more and more localised. The amount of non-English speaking users is steadily growing and these people expect contents they can read and understand. Language, however, is not the only aspect that provides diversity within the user base: culture is also another critical factor. The world has many cultures and they all present significant differences which affect people's behavior and perception. Being aware of the existence of such differences and able to cope with them is called *cultural sensitivity*. It is, indeed, rather common in real life to face situations where there is a dominant and one or more secondary cultures [153]. For instance, in the U.S. the European American is the dominant culture whereas Hispanic, African American and Chinese cultures are all secondary. Cultural sensitivity implies that both groups understand and respect each others characteristics. Achieving cultural sensitivity is always a challenge and even more so in environments where the dominant culture is the one people are expected to adopt. The Internet nowadays is an outstanding example of an environment wherein cultural sensitivity is yet to achieve. The dominant culture of the Internet is the Anglo-American one, with all the other ones confined to the role of secondary cultures, and Web users are expected to conform to some extent to that culture to fully enjoy the contents and services offered by the Web platform. However, the growing number of non-English speakers using the Internet on daily basis is steadily growing as more and more developing countries are gaining a better Web access. Content published on the Web, more than ever, can now be consumed by people from all around the world, thus content providers should be as much cultural sensitive as possible to avoid being considered offensive, inappropriate, or needlessly aggressive. Even service providers, such as search engines, should become more cultural sensitive, since the "one size fits all" approach currently adopted by most of them[1] is

[1]Several service provides adopted Adaptive Personalisation techniques over the past few years, however this feature often has a limited coverage, mostly focusing on English language, or is based upon Web usage mining techniques that do not take into account cultural differences among monitored users.

highly unlikely to satisfy equally users coming from different cultures.

To provide a better access to the multitude of content published on the Internet, fulfilling the goals and expectations of users speaking different languages and belonging to different cultures, intelligent tools are needed. In particular, being able to understand the textual content of different languages and formats is becoming a more and more desirable feature for several practical tasks such as information retrieval, content placement, personalisation, and Web usage monitoring. While big efforts have been made to understand English content, other languages have received less coverage over recent years. On the other hand, the quantity of localised, i.e. non-English, content on the Internet has significantly risen[2] and will probably be rising even more in the next years due to the growing access to the Web of developing countries.

1.1 Language and Understanding

It is well known and widely accepted in linguistics that the language and its usage can be split into several components [50, 20, 134, 62]. The five main components of language are phonemes, morphemes, lexemes, syntax, and context. Along with grammar, semantics, and pragmatics, these components work together to create meaningful communication among individuals[3].

In the case of written communication, which is prominent on the Internet, the phonetic part of the language is absent, however the remaining four components are still present and can be arranged in a hierarchical order, as shown in Figure 1.1. Morphemes are the building blocks of lexemes, which are arranged to create sentences according to the syntax which provides semantics. Finally, the context in which sentences are put determines the discourse pragmatics. The notion of context is, however extremely broad since it may include, but is not limited to, previous discourse, background and common knowledge, and non-verbal communication.

This decomposed view of the language is particularly useful for the purpose of machine understanding of the language since it provides a succinct framework to identify the degree of textual comprehension of automated tools. As we will broadly illustrate in Chapter 2, the lower the level of abstraction the easier it is its processing by a machine. As a matter of fact, there already exist several Natural Language Processing tools that operate with human-like precision in parsing and decomposing words, phrases, and sentences. Examples of such tools are Lexers, Stemmers, Part of Speech Taggers, and Spellcheckers. The upper part of the linguistic stack shown in Figure 1.1, however, is still a research subject, with large progress being made over

[2]W3Techs: Usage of content languages for websites - `http://w3techs.com/technologies`

[3]Extracted from: Boundless. "The Structure of Language" Boundless Psychology Boundless (visited on 26 May 2016) - `https://www.boundless.com/psychology/textbooks/boundless-psychology-textbook/language-10/introduction-to-language-60/the-structure-of-language-234-12769/`

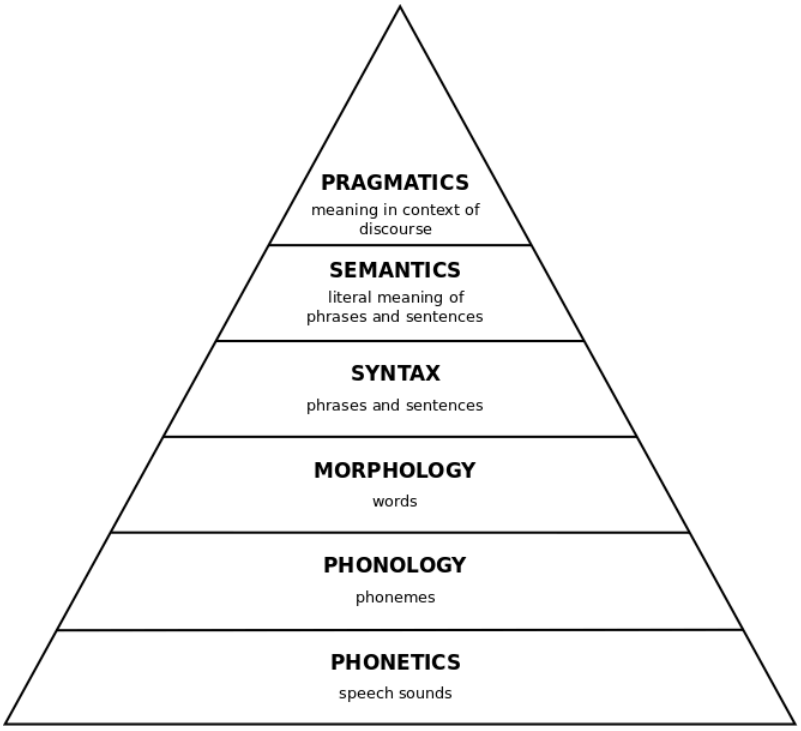

Figure 1.1: Relationship between types of linguistic units.

the past 15 years, but with performances still far from being human-like. Semantics, in particular, have been thoroughly explored over recent years with the support of the Linked Data community, however, tasks such as Entity Linking, i.e. identifying references to entities in a knowledge base within an unstructured text, are still hard to automate with satisfactory accuracy.

Most of these research activities, however, share a common liability: they have been tailored on a single language, mostly English. While one could argue that general semantics and pragmatics are independent from the language, they still lay on the top of it and are heavily influenced by cultural background and other local factors. State of the art systems, therefore, lack multilingual and cross-cultural components, with very few notable exceptions. The world, however, is multi-lingual and multi-cultural: distances have suddenly shrunk, the Internet is becoming more and more localised, and people nowadays expect localised services as well. The need for multilingual and cross-cultural language analysis tool is, in our opinion, one that should be addressed as soon as possible.

1.2 Considered languages

To investigate the challenges of multi-linguality and cross-culturality we will consider in the following three languages, each with unique characteristics: English, Italian,

and Arabic.

English is considered the most widespread language on the Internet. Despite the fact it is impossible to assess the actual distribution of languages on the Internet [168] due to its size, English is constantly the most widespread language found in random samplings performed with crawling techniques, such as the notorious Common Crawl corpus [157] and several others. Moreover, most research work in the Natural Language Processing field is based on this language that is also regarded as the current *lingua franca* of the international scientific community. Linguistically speaking, English is a West Germanic language characterised by little inflection, a fairly fixed Subject-Verb-Object word order and a complex syntax [86]. The fairly simple morphology of the English language, paired with the abundance of linguistic resources available makes it an ideal proving ground for most NLP and Information Extraction applications.

Italian is a Western language as English, but it is a Romance Language, most notably it is considered the closest spoken language to ancient Latin [67].With respect to English, Italian has a much richer morphology, a more complex grammar which showcases all typical constructs of Romance languages, such as genders, and inflection, moreover the order of words in the phrase is relatively free compared to most European languages and it allows null subject sentences [31].

Arabic, on the other hand, is a Central Semitic language closely related to Hebrew and Aramaic. The authors of [57] and [69] highlight the most notable features of Arabic from an NLP point of view. Its main characteristics are the absence of letter capitalisation, changing letters according to their position, minimal use of punctuation, ambiguity due to the presence of homographs, complex internal structure of sentences that makes their interpretation highly context-dependant. Furthermore, Arabic is a strongly agglutinative language, like German, allowing the aggregation of whole phrases in one word, and also allows the dropping of subject pronouns. Arabic is also a relatively free word order language, implying that phrase patterns may frequently vary. Finally, Arabic presents *diglossia*, which means that there exist multiple variants of the language (Classic Arabic, Modern Standard Arabic (MSA), local dialects, ...). All these characteristics make Arabic unique and suggest that techniques that do not take them into account may achieve poor performance to the eyes of a proficient Arabic speaker.

These three languages, with their distinctive characteristics, provide a rich and comprehensive case study to the extents of developing truly localised Artificial Intelligence techniques. Moreover the total number of native speakers of these three language is somewhere around 1.2 billion people, making such a case study also relevant from an application point of view since the number of potential users interested in is large.

1.3 Thesis outline

Processing different languages such as the ones described in the previous section implies overcoming a series of technical and methodological challenges. We can pinpoint some of the most notable and pressing ones that we will address in this thesis:

- The need for an abstraction over syntax: while it is common sense that some assumptions may yield over several languages (especially the ones concerning semantics) the morphological aspect of a language must be taken into consideration. Truly multilingual approaches should, in our opinion, provide and abstraction over syntax and morphology to be filled with the appropriate tools for each language.

- Robustness to style nuances: there exist many types of text (Narrative, Legal, Technical, and many more) each one with its unique features. Moreover, these features may change from culture to culture. Ideal multilingual and cross-cultural tools should provide help to tackle this aspect of communication.

- Commonsense knowledge differences: different communities tend to have different background knowledge; this is particularly true when the consider communities are separated by large distances and cultural barriers. For instance, the average American has a radically different idea of the notion of "healthy food" from the average Italian. These cultural differences cannot be ignored to grasp the real meaning of texts, especially on a vivid and dynamic environment such as the Internet.

- Acceptable computational times: deep text analysis is a very demanding task from a computational point of view, especially when it comes to resolve its semantics. Most approaches presented in the field of semantics rely on description logics or similar formalism, however reasoning on such structures is a nontrivial problem in complexity. The implied computational times are not acceptable for most practical applications, and therefore optimisations and/or simplifications of these models are needed.

The rest of this thesis is organised as follows: in Chapter 2 we will briefly describe some notable work already published in the literature, then, in Chapter 3 with reference to a specific NLP task, Keyphrase Extraction, we will present a knowledge oriented framework to describe such an activity and introduce a multilingual implementation of the said task, in Chapter 4 we will address the problem of representing background knowledge and propose an approach able to cope with different cultural backgrounds and with the computational demands of real-world application, finally, in Chapter 5 we present our final remarks and suggest some future research directions paved by this work.

2

Text Mining

In this chapter we will introduce previous work done in the field of automatic Information Extraction from unstructured text and Knowledge Representation. We will focus at first on the problem of extracting *relevant* information from unstructured text, introducing the task of Keyphrase Extraction, than we will introduce other Information Extraction problems, namely Named Entity Recognition, Word Sense Disambiguation, and Named Entity Linking, finally we will provide an overview of Knowledge Representation techniques.

In this chapter we will also stress two critical issues of state of the art techniques: the lack of multilinguality, i.e. the ability of a technique to scale over different languages, and multiculturality, i.e. the capability of a knowledge representation technique to cope with cultural differences and subjective visions over a given topic.

Some of the results presented in this state of the art survey are also published in our previous work [49, 47, 42, 77, 48, 46]

2.1 Keyphrase Extraction

Citing SEMEVAL 2010's Keyphrase Extraction task description, Keyphrases (herein KPs) are words that capture the main topic of the document, therefore they can be seen as special n-grams[1] that are relevant to the extents of describing, summarising, or indexing an arbitrary long text. The problem of extracting KPs from natural language documents has already been investigated by several scholars and many different approaches have been proposed.

All known techniques can be substantially broken down into two steps: the candidate generation phase where all plausible keyphrases are spotted in the text, and the candidate selection phase where the relevance of all candidate keyphrases is assessed and the final ones are subsequently selected.

In an effort of organizing the wide range of approaches that has been proposed in the literature, the authors of [177] identify four types of keyphrase extraction strategies:

[1] a contiguous sub-sequence of n items of a given sequence.

- *Simple Statistical Approaches*: these techniques assume that statistical information is enough to identify keywords and KPs, thus they are generally simple and unsupervised; the most widespread statistical approaches consider word frequency, TF-IDF or word co-occurrence [110, 145, 89]. It is important to note how TF-IDF based methods require a closed document corpora in order to evaluate inverse frequencies, therefore they are not suitable for an open world scenario, where new items can be included in the corpora at any time.

- *Linguistic Approaches*: these techniques rely on linguistic knowledge to identify KPs. Proposed methods include lexical analysis [6], syntactic analysis [56], and discourse analysis [87, 90].

- *Machine Learning Approaches*: since KP extraction can be seen as a classification task (each KP can be considered a class to which the document belongs), machine learning techniques can be used as well [60], [163] and [79]. The usage of Naive Bayes, SVM and other supervised learning strategies has been widely discussed and applied in systems such as KEA [175], Maui [112], LAKE [36], and GenEx [163].

- *Other Approaches*: other strategies exist which do not fit into one of the above categories and most of the times they are hybrid approaches combining two or more of the above techniques [45, 68]. Among others, heuristic approaches based on knowledge-based criteria [99], and meta-knowledge over the domain [35] have been proposed.

At first glance, using TF-IDF metric [148] as a method for extracting keyphrases, could be considered a generally reliable solution; the main issues, however, are the need of a corpus of documents (that is not always available), the necessity of defining a threshold of relevance above which n-grams can be considered relevant, and moreover, TF-IDF simply does not take into account the internal structure of a document and its properties not exploiting useful features that are then wasted.

Many different features have been presented in the literature; a detailed list of of which is presented in Table 2.1.

2.1.1 Multilinguality

The problem of defining multi-language techniques, though somewhat neglected, has been discussed as well in the literature. Some authors, indeed, already addressed some fundamental issues and proposed some working systems, however most of the proposed approaches consist in a minor reworking of techniques conceived for the English language.

A multilingual approach towards sentence extraction for summarization purposes based on a machine learning approach can be found in [98]. The authors of [136] introduce a multilingual KP extraction system exploiting a statistical approach based

Table 2.1: Features used in literature to perform KP extraction.

Feature	Meaning	Type	Used in
Number of words	Number of words in the candidate keyphrase	D	[163, 71, 100, 83]
Number of characters	Number of characters in the candidate keyphrase	D	[108]
Candidate first occurrence (or depth)	First occurrence of the stemmed phrase in the document, counting with words	D	[163, 175, 60, 79, 71, 129, 83, 100, 141]
Candidate last occurrence	Last occurrence of the stemmed phrase in the document, counting with words	D	[83, 141]
Candidate stem first occurrence	First occurrence of a stemmed word of the candidate, counting with words	D	[163]
Normalized phrase frequency (TF)	Frequency of the stemmed phrase in the document (TF)	D	[163, 79, 71, 141]
Relative length	Number of characters of the candidate	D	[163]
Proper noun flag	Candidate is a proper noun	D	[163]
Final adjective flag	Candidate ends with an adjective	D	[163]
Verb flag	Candidate contains a known verb	D	[163]
Acronym flag	Candidate is an acronym	D	[83, 129]
tf-idf over corpus	TF-IDF of the candidate in the corpus	C	[175, 60, 79, 100, 129, 83]
keyphrase frequency	frequency of the candidate as a keyphrase in a corpus	C	[175, 60, 164, 100]
candidate frequency	frequency of the candidate in the corpus	C	[79]
POS sequence	sequence of the POS tags of candidate	D	[79], [129], [108]
Distribution of the POS sequence	distribution of the POS tag sequence of candidate in the corpus	C	[83]
number of named entities	number of named entities in the candidate	D	[108]
number of capital letters	used to identify acronyms	D	[108]
IDF over document	inverse document frequency	D	[71]
Variant of TF-IDF - 1	logTFIF - see [71]	C	[71]
First sentence	First occurrence of the phrase in the document, counting with sentences	D	[71]
Head frequency	Number of occurrences of the candidate in the first quarter of the document	D	[71]
Average sentence length	average length of the sentences that contain a term of the candidate	D	[71]
Substring frequencies sum	sum of the term frequency of all the words that compose the candidate	D	[71]
Generalized Dice coefficient	see [71] or [100]	D	[71], [100], [83]
Maximum likelihood estimate	estimation of the probability of finding the candidate in the document	D	[71]
Kullback-Leibler divergence	see [71]	C	[71]
Document phrase maximality index (DPM)	see [71]	D	[71]
DPM X TF-IDF	self-explanatory	C	[71]
Variant of TF-IDF - 2	TF-IDF of the candidate / TF-IDF of its most important word (see [71])	C	[71]
k-means of the position	see [71]	C	[71]
GRISP presence	presence in the GRISP database (see [100])	E	[100]
Wikipedia keyphraseness	probability of the candidate to be an anchor in Wikipedia	E	[100]
Title presence	Presence of the candidate in the title	D	[100]
Abstract presence	Presence of the candidate in the abstract	D	[100]
Introduction presence	Presence of the candidate in the introduction	D	[100]
Section title presence	Presence of the candidate in a title of a section	D	[100]
Conclusion presence	Presence of the candidate in the conclusions	D	[100]
Reference or book title presence	Presence of the candidate in at least one reference or book title	D	[100]
Variant of TF-IDF - 3	TF includes the TF of substrings of the candidate	C	[83]
Variant of TF-IDF - 4	TF of substrings of the candidate without the TF of the candidate	C	[83]
Variant of TF-IDF - 5	TF normalized by candidate types (noun phrases vs simplex words vs...)	C	[83]
Variant of TF-IDF - 6	TF normalized by candidate types as a separate feature (not clear)	C	[83]
Variant of TF-IDF - 7	IDF using Google n-grams	E	[83]
Section information	Weight the candidate based on its location (abstract, title, ...)	D	[129], [83]
Section TF	TF of the candidate in key sections	D	[83]
Candidate co-occurrence	Number of sections in which the candidates co-occur	D	[83]
TF Occurrence in titles	Occurrence in the CiteSeer title collection as substring of a title	E	[83]
Occurence in titles	TF of the candidate in the CiteSeer title collection as substring of a title	E	[83]
Semantic similarity - 1	contextual similarity among candidates	D	[83]
Semantic similarity - 2	semantic similarity among candidates using external knowledge	E	[164] (using a search engine)
Variant of Dice coefficient - 1	normalized TF by candidate types (noun phrases vs simplex words...)	D	[83]
Variant of Dice coefficient - 2	weighting by candidate types (noun phrases vs simplex words...)	D	[83]
Variant of Dice coefficient - 3	normalized TF and weighting by candidate types (noun phrases vs simplex words...)	D	[83]
Suffix sequence	Sequence of the suffixes of the words that from the candidate	D	[129], [83]
Semantic similarity - 3	Co-occurrence based similarity	D	[110]
Variant of TF-IDF - 8	Probability-based (see 3.4 of [164])	C	[164] (using a search engine)
First sentence	First occurrence of the phrase in the document, counting with sentences	D	[9]
Last sentence	Last occurrence of the phrase in the document, counting with sentences	D	[9]
Lifespan on words	Difference between the last and first appearance in the document	D	[141]
Lifespan on sentences	Difference between the last and first appearance in the document	D	[9]
Wikiflag	Presence of the candidate as a Wikipedia page title or surface (e.g. Big Blue vs IBM)	E	[45]
Noun value	Number of nouns in the candidate	D	[141], [45], [9]

on word frequency and a reference corpus in 11 different European languages, including Italian. The performance of such system, however, relies on the quality of the reference corpus since phrases not included in the corpus will never be extracted from the text. Moreover, its accuracy proved to be highly variable over the 11 considered languages and overall poor. The authors of [53] propose a more sophisticated approach based on a set of heuristic rules for identifying a set of potentially good candidate KPs; candidate KPs are then selected according to a TF-IDF based score metric. The system exploits two language dependant resources: a stopwords list and a stemmer. Upon a suitable substitution of such language dependant resources, the system proved to perform well in different languages.

In the next chapters we will focus on two languages in particular: Arabic and Italian. Keyphrase extraction from Italian texts has received little attention. The authors of [59] propose *TagMe*, a system whose purpose is to annotate documents with hyperlinks to Wikipedia pages by identifying *anchors* in the text. The task of identifying text anchors can be seen as a naive KP extraction technique and it is capable to identify and propose KPs only if they are also in Wikipedia. The system by [136], previously mentioned, is also capable of extracting KPs from Italian text, however it features a very limited accuracy.

Keyphrase extraction for the Arabic language has not received much consideration basically adapting techniques developed for western languages. A prime example of this situation is given by KP Miner [53], which leverages *TF-IDF* and it is built using an unsupervised approach yielding satisfactory results in both Arabic and English. KP Miner, although exploiting purely statistical techniques, performs its task effectively and it is considered the de facto standard to which to compare alternative systems. The authors of [54] employ a supervised approach for the selection of KPs in accordance with the linguistic features obtained through a Part-Of-Speech (herein POS) tagging. The inclusion of such linguistic features greatly increases the accuracy of the system similarly to what happens in the English language [79]. In [2] the Multi-Word-Expressions are introduced which, although with a slightly different definition, appear similar to the KPs. The authors compare different techniques to extract them from Arabic documents exploiting Wikipedia, Google Translate, and distributional features of the text corpus under analysis. The approach presented in [4] includes a "cleaning" phase which removes candidate KPs according to linguistic knowledge, and then groups terms into equivalence classes according to their roots. The evaluation is then performed on an ad-hoc built human annotated test corpus upon which the authors claim to achieve significantly higher precision and recall than KP Miner.

2.2 More Information Extraction Tasks

Even though the use of keyphrases is of great help for the representation of texts and their summarisation, they own little intrinsic semantic value since, for instance,

the same KP may refer to different entities. The embedding of a semantic layer in the information extraction process is a critical step for the inclusion of a semantic level in applications that will use its functionality. NLP tasks such as Named Entity Recognition, Word Sense Disambiguation, and Named Entity Linking could help to have a better understanding of a natural language text by classifying token of text or identifying them in knowledge bases, ontologies, dictionaries, or gazetteers.

2.2.1 Named Entity Recognition

Named-Entity Recognition (NER) (also known as *entity extraction*, *entity identification*, and *entity chunking*) is a subtask of Information Extraction that seeks to find text strings representing entities and concepts in a natural language text [123], which can be seen as a classification problem where token of text must be labelled with the class of the entity they refer to. The most common NER usage is the detection of a limited number of classes within a collection of documents. As an example, in *SemEval 2017*'s tenth task[2], three classes ("Task", "Process", and "Material") must be spotted in a text corpus built by collecting scientifc papers' abstracts. NER tasks can also be seen as sequential prediction problems, and are commonly addressed employing distributional semantics and leveraging sequence tagging methods like sequential applications of *Perceptron*, *Hidden Markov Models* (HMM), or *Conditional Random Fields* (CRF). CRFs, in particular, have emerged in the last few years as the *de facto standard*, especially in the biomedical research field [97, 91, 128]. From a theoretical point of view, the problem can be formalized as follows: let $x = (x_1, ..., x_n)$ be an input sequence and $y = (y_1, ..., y_n)$ be the corresponding output sequence, the sequential prediction problem is to compute the probabilities $P(y_i|x_{i-k}...x_{i+l}, y_{i-m}...y_{i-1})$ where k, l, and m are small integer numbers, to achieve tractable inference and prevent overfitting [142]. In other words the text is processed with a sliding window that considers k words backward, l words afterwords and the last m predictions generated by the tagger. Large sets of features are usually employed in NER to compute such conditional probability.

The authors of [161] identified three different classes of features generally used in NER:

- *Local knowledge features*: features that can be obtained from the word they encode. They include capitalisation, the presence of specific suffixes, prefixes, or special characters, and the presence of sub-tokens like the ones identified by hyphenation (e.g. the word "high-tech" can be split into the two sub-tokens "high" and "tech").

- *External knowledge features*: features that, to be gained, require some background knowledge, such as linguistic or encyclopedic, and that cannot be inferred directly from the text. They include POS tagging, word or phrase

[2] `https://scienceie.github.io/`

clustering analysis over a reference text corpus, and any information collected by matching the examined word against gazetteers, thesauri, or ontologies.

- *Non-local dependency features*: features assembled taking into account the hypothesis that the context in which a word is inserted shapes its meaning, and therefore they try to represent the surrounding phrase, sentence, or discourse. They include the number of times the examined word appears in a window, the presence of other significant words within a certain window, context aggregation [26], and a possible preliminary classification given by another sequential tagging algorithm [88].

The authors of [142] present convincing clues that all these three kinds of features grant effective results in NER tasks and should, therefore, be all considered for the design of feature sets to be used to train sequence tagging algorithms. Common NER systems, however, are typically trained to identify a limited number of different entities classes in the text (e.g. nations, companies, people, places). To achieve better results, however, it is advisable for several NER applications to extend the classification to a much larger and fine-grained number of classes.

2.2.2 Word Sense Disambiguation and Named Entity Linking

Word Sense Disambiguation (WSD) can be defined as the task of selecting the right sense for a word within a given context [119]. In this domain, the matched string is commonly referred to as the *surface form* of its corresponding meaning. The main difference from the NER task is that the latter associates a string with the corresponding class, while WSD associates a string with a specific item in a dictionary, such as a Wordnet[3] synset. The string-meaning association can be even more accurate by associating string tokens with the corresponding node of a knowledge base (e.g Wikipedia or its Linked Data equivalent DBpedia [93]). This task is typically denominated *Named Entity Linking* (NEL) [70] in general or *Wikification* when the target knowledge base is Wikipedia [23, 32]. Both tasks, WSD and NEL, are usually performed in two steps:

- *Candidate anchor search*: The text is scanned and all the tokens that can designate entities are detected. In this step, heuristic-based search techniques or vast dictionaries and gazetteers are commonly used.

- *Entity selection*: Among all potential candidate tokens, those that actually refer to an entity are identified, linking them, if it is possible, to the entity itself. Many techniques have been developed to evaluate the plausibility that a string is referring to an entity and to disambiguate polysemic words through the context in which they appear.

[3] `http://wordnet.princeton.edu`

The candidate anchor search phase can be further decomposed into two steps: (i) the text tokenization, concerning the detection of sentence boundaries and the possible misspell or miss-capitalization of the words, and (ii) the detection of surface forms within the chunked text. This last point can be addressed in different ways. The most widely used approach consists in matching the surface forms against dictionaries or gazetteers [23], rule-based matching driven by linguistic hypothesis, or the adoption of NER systems to detect specific classes of entities. State of the art techniques include the usage of coreference resolution to map short surface forms, such as acronyms and abbreviations, to longer surface forms with the same label [72], the integration of the aforementioned strategies into a synergic search pipeline [32], and the use of fuzzy matching algorithms [94, 169].

The second phase (Entity selection) can be addressed either by using distribution semantics techniques or taking advantage of the ontological structure of a target knowledge base. Approaches based on distributional semantics rely on a reference corpus of annotated texts and they are trained to recognize surface-sense associations taking into account the context in which the surface is included. In the Wikification case, Wikipedia articles are considered as a corpus of annotated texts in which entities are described by the presence of hyperlinks to other articles of the knowledge base. A representation of context is analysed for each surface-sense pair, taking into account the co-occurrences of words in the training corpus. When a non-annotated surface form needs to be assigned to an entity or to a meaning, the context in which it is inserted is evaluated. All possible assignments are considered and the one with the highest similarity index based on the context is then assigned [32]. This approach is used by the vast majority of Wikification systems such as *TagMe* [59] and *DBpedia Spotlight* [114]. On the other hand, network-based approaches rely on the internal structure of a knowledge base or, in the case of Wikipedia, on its internal linked structure that forms a dense and navigable network of interlinked documents. Exploiting a large enough network, it is possible to take advantage of its structure to effectively accomplish disambiguation and entity selection. The most appropriate surface-sense matching pair can be determined by finding the one pair that minimises the distance with the already grounded terms. This approach makes extensive use of clustering techniques and graph search algorithms. The authors of [166] exploit a reference ontology to disambiguate concepts, computing the degrees of separation between candidate items. The authors of [119, 118], instead, join Wordnet with Wikipedia, thus obtaining a much larger knowledge base than the one used by plain Wikification systems, and they implement WSD relying on a random walk with a restart on minimum support graphs.

Even though they share similar techniques, it is important to point out, how WSD and NEL are guided by different hypotheses. In particular, we can identify the following three main differences [70]:

- Nature of the External Knowledge source: WSD systems are based on purely linguistic assets such as dictionaries and lexicons, while NEL systems rely on

domain knowledge provided by domain ontologies.

- Completeness of the Knowledge source: WSD systems assume their knowledge base to be complete, i.e. the lack of a potential association for a candidate surface implies the absence of a meaning for that word. NEL systems, on the other hand, assume their knowledge to be incomplete, i.e. every candidate surface form should be considered as an entity, even when it is not possible to find an association between it and an entity of the knowledge base [23, 111]. This latter hypothesis is often ignored by Wikification systems; in this respect, they lay halfway between WSD and NEL applications[4].

- Candidate search: named entity mentions are more various than lexical mentions in WSD. This is caused by the wide variety of abbreviations, synonyms, and paraphrases that are encountered when dealing with domain-specific jargon and the fact that entities defined by long and complex words are usually referred to in different ways within the same text [170]. The candidate search phase, so, can be considered more challenging in NEL systems design. Moreover, there are indications in the literature that advanced candidate search techniques, like query expansion based on coreference resolution, have a major impact on the accuracy of NEL systems [70].

Apart from the distinctions listed above, NER, WSD, and NEL systems share remarkable conceptual overlaps. In fact, they are often built on similar technologies and assumptions and, moreover, they can be employed in the same tasks and applications. In the research paper [143], the authors propose the *NERD* framework which aims at addressing the possible overlaps between these three tasks and provide a development environment for building this kind of applications.

2.3 Knowledge Representation

Up to this point of the discussion, we dealt with techniques to extract keyphrases from textual documents and link these KPs to a set of labels that represent items or categories taken from a trustworthy knowledge base, thus representing entities. Nevertheless, for several tasks this information is not enough and *Knowledge Representation* is needed to associate to KPs and entity links a background allowing an artificial intelligence system to perform some kind of *reasoning*. The most straightforward way of representing knowledge is using formal logics, however several alternative approaches are viable as well. In this section we will survey Linked Open Data and their related technologies as formal representations and semantic distances as distributional knowledge representation.

[4] As a matter of fact, some authors consider Wikification as the bridge between WSD and NEL [119]

2.3.1 Formal Knowledge Representation

Historically, automated reasoning has always been associated with logic, and as a matter of fact, most classical work in the field of Artificial Intelligence relies upon First Order or Description logics to some extent. Formal knowledge representation formats have been implemented and represented in several different ways over the past 50 years of Knowledge Representation research, notable examples are Semantic Networks [156], Horn Clauses [78], Conceptual Graphs [155], and Frames [116]. Nowadays, the most widespread format to represent formal knowledge are *Linked Data*, which are a Web-oriented implementation of Semantic Networks. Linked Data are associated with the Semantic Web, being its preferred format to represent metadata, domain knowledge, and business logic as well. Some Linked Data are distributed with open licences, allowing researchers and practitioners all over the world to contribute them and to exploit them as knowledge bases for novel applications. Such Linked Data are commonly referred to as *Linked Open Data* (LOD) and over the past 20 years they gradually formed the so-called *LOD cloud* which is an interconnected collection of structured data publicly available on the Web. As of January 2017, the LOD cloud includes 1139 interlinked datasets and several billions of triples[5].

The Semantic Web stack

Linked Data are not a single technology, rather they consist in a stack of technologies known as the *Semantic Web Stack*. More precisely, Linked Data are built on a subset of the Semantic Web stack. The full stack includes:

- Web Platform: also known as the *level zero*, it includes all the basic technologies of the Web, such as the UTF-8 character encoding, the URIs, the HTTP protocol, and all the other common Web technologies and infrastructures.
- The Syntax: this level includes semi-structured data formats such as XML. JSON, and similar ones that are used to serialise Linked Data.
- Data Model: this level consists of the *Resource Description Framework* (RDF) data model, which provides a data exchange format abstracting over the actual serialisation of data. In RDF the atomic unit of data is the *triple* which consists in a binary predicate, usually represented in the form *subject-predicate-object*.
- Domain Model: this level provides domain modelling capabilities which include vocabulary specification, domain constraints, and axioms. This level can be provided by several technologies such as RDFS, OWL, SKOS, RIF, and many other, often used in an ensemble. OWL DL is the recommended technology, and it implements a $\mathcal{SHOIN}^{(\mathcal{D})}$ description logic.

[5] for the current state of the LOD cloud we address the curious reader to `http://lod-cloud.net/`

- Query: this level includes technologies to query the data built using the technologies provided by the previous layers of the stack. The W3C recommendation query language is SPARQL 1.1, but several alternatives are available as well.

- Logic and Proof: this layer provides reasoning over data built with the previous layers of the stack. Being the Data Model and Domain Model layers based on Description Logics, plenty of reasoning tools are applicable.

- Trust: this layer provides meta-information to track provenance, authorship, and trust in general. This layer is critical for Linked Data contributed by many sources, since a single inconsistency could break gigabytes of data, moreover this layer is required to provide Data Citation. Unfortunately, this layer cannot be considered fully implemented and many of its issues still are open research problems.

- Application: the topmost level of the stack, it includes applications built on top of the lower levels of the stack.

- Security: while it is not a proper layer of the stack, security is a service that can be included at any level of the stack with the usage of cryptography.

This stack over the years has grown more and more complex, to the point of not being anymore a proper stack, since the relationships between the different layers are not linear anymore and different applications can interact with many levels of the stack. Figure 2.1 shows the intricate relationship among the technologies that compose the Semantic Web Stack. Linked Data use only the first four levels of the stack, with arguably the fourth one, Domain Model, being the most interesting to the extents of knowledge representation since it encodes domain assumptions, business logic, and often complex constraints. Domain models built with Semantic Web technologies can be of three kinds:

- Vocabularies: they consist in an enumeration of domain concepts and properties, with no relationships or constraints. Languages such as SKOS are meant primarily for vocabulary specification.

- Taxonomies: vocabularies with a single hierarchical relation. Typically they are built with RDFS.

- Ontologies: vocabularies with multiple relations, not necessarily taxonomic, and constraints used to encode business logic.

In principle ontologies may not be computable, however the Semantic Web stack offers tools that limit the expressive power of the modelling language by the design,

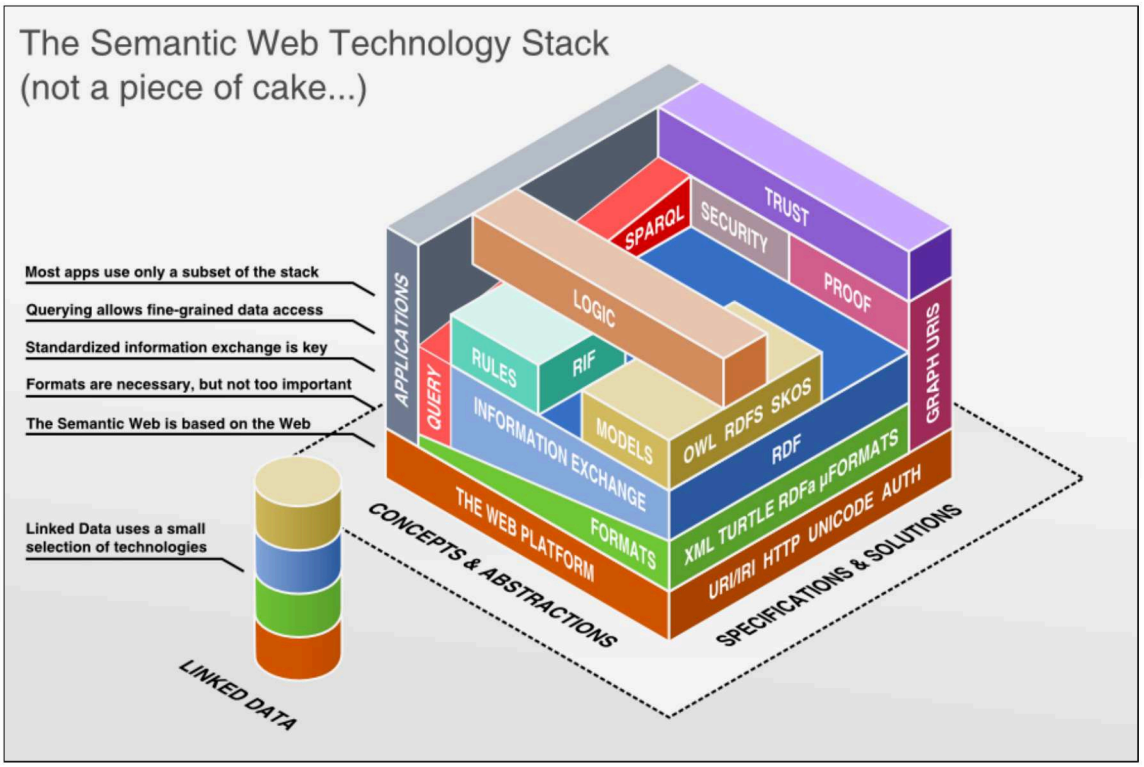

Figure 2.1: The Semantic Web Technology Stack (Figure by Benjamin Nowack - CC BY 3.0).

retaining computability[6]. The most critical aspect of ontologies is, however, the huge workload their development implies: formalising an application domain, even if narrow, is a knowledge-intensive task that requires time and expertise, therefore the development of an ontology is ludicrously expansive. To address this issue, substantial effort was put over the last years into automatic ontology construction.

Domain Ontology Extraction

Building ontologies in an automated way is a seductive idea given the high costs implied by building them manually and notorious issues related to subjectivity, confirmation bias, and cultural biases implied by manual ontology design. The Internet and Digital Libraries offer a near-unlimited amount of text to mine to extract entities, classes, and relationships. Distributional information over words and sentences can identify relevant entities and semantically related ones, as we will discuss in the next paragraph, however, qualifying relationships among them is still a complex problem. From the literature, four different strategies emerge that can

[6] An improper combination of different modelling languages allowed by the Semantic Web stack may still result in undecidable ontologies. This is a severe and open issue, however we will not discuss it in this thesis.

be used to detect the relationship between terms:

- Natural Language Processing techniques: relationships are inferred from the structure of text wherein entities are mentioned [29]. A prime example of NLP technique is *Lexico-Syntactic Pattern Extraction*: relationships are deduced finding linguistic patterns in the text, like "and other ...", "in the likes of ...", and so on [76]. These methodologies require an extensive text corpus and cannot be applied to meta-information since they need textual context.

- Clustering techniques: to identify some relationships, such as synonymy or taxonomic ones, entities can be clustered according to the different contexts wherein they can be found [105]. To identify taxonomic relationships, various systems such as the *TaxGen framework*, rely essentially on hierarchical clustering over huge text corpora [120].

- Conditional Probability-based techniques: for each entity under examination, a conditional probability of being connected with the ones already present in the considered data set is computed. Taxonomic relationships are then inferred from such probabilities. The most popular approach to estimate these relationships is using the subsumption method [149], nevertheless, numerous alternatives have been proposed, including considering second order co-occurrences computed with a variant of the *Page-Rank algorithm* [51].

- Graph-based techniques: a complex network is created beginning with a simple origin ontology and then combining other ontologies and entities obtained through text analysis or from additional metadata. Relationships and entities to be incorporated in the final ontology are then identified using spreading activation [176]. These methods are commonly employed to extract relevant subsets of larger knowledge bases and ontologies [131].

An instance of domain ontology extraction system designed for data access in the scholarly domain is the *Klink-2* algorithm discussed in [132]. Klink-2 identifies three relationships defined by the *BIBO* ontology[7]: *skos:broaderGeneric*, *contributesTo*, and *relatedEquivalent*, with the last two being subproperties of *skos:related*. Several methods are employed to spot these relationships: *relatedEquivalent* is inferred using hierarchical clustering, while the hierarchical relationships *skos:broaderGeneric* and *contributesTo* are inferred with a modification of the subsumption method, joined with domain knowledge that exploits temporal information to recognize narrower and broader topics. Klink-2 takes advantage of a wide variety of data, including textual data, semantic information collected from the Linked Open Data cloud, and scientific publications metadata. When enough data are provided, Klink-2 can build accurate topic taxonomies merging these multiple kinds of knowledge.

[7] `http://purl.org/ontology/bibo/`

Criticalities of Linked Open Data and Ontologies

As hinted in the previous paragraphs, despite over twenty years of research on Semantic Web technologies, many issues still remain open problems. Letting aside the countless challenges posed by ontology engineering in general and by the intrinsic computational complexity of Description Logics involved in these technologies, many other problems limit the field usage of such technologies.

First and foremost, ontologies, being formalisations of application domains handmade by domain experts, are intrinsically subjective: their design choices are driven by domain assumptions made by their authors, personal understanding of the modelled domain, and pragmatic factors such as interoperability with other knowledge bases, technical requirements, and willingness to expose business logic. As a direct consequence, a huge number of ontologies has been proposed, each one of them representing a unique conceptualization of a given task or domain, and often the same domain is conceptualized in different ways by different ontologies. Linked Data are conformed to convenient ontologies, resulting in a wide range of different ontologies being actively used on the Web.

To combine data conform to different, but related, ontologies, an *ontology alignment* must be specified. An ontology alignment can be defined as any formal representation of a set of relations between two ontologies [55, 17]. From a practical perspective, alignments consist of a set of bridge axioms between two ontologies and then alignments themselves can be considered ontologies. Creating ontological alignments is a very challenging and knowledge intensive task, that involves a vast variety of domain experts. These difficulties imply that they are usually built completely by hand. With the fast rise of the number of ontologies due, among other things, to the rapid growth of the Web of Data, alignments between ontologies have become of extreme value as they are a key component of any data integration activity. In recent years there has been a fairly good effort by the Semantic Web community to suggest methods to automate the process of ontology alignment [27, 154]. Most of the proposed techniques try to identify shared concepts among different ontologies making use of the content of large natural language text corpus [127, 24, 104], however such systems tend to exploit naive statistic techniques to spot entity references and, therefore, detect synonyms.

Another critical aspect of Linked Data field usage is the lack of proper version control, authorship verification, and trust mechanisms. As introduced in Paragraph 2.3.1, the Trust level of the Semantic Web stack is still a research topic and there not exist a comprehensive W3C recommendation on the subject. While most other artifacts, such as source code, database records, and multimedia items can be managed with well known best practices that address these problems, Linked Data cannot. Some triplestores (i.e. repositories tailored for RDF triples) are backed by a relational database to address this issue, but this solution is more akin to a workaround.

A final, and up to now not mentioned issue with Linked Data is related to data themselves: once an ontology is designed, it still has to be populated with

individuals[8] and their properties. This task can be as hard as ontology design itself, especially when considering particularly complex ontologies aligned with several other ones. Let us take as an example DBpedia, which is one of the most famous LOD dataset available: since it is aligned with several other ontologies, it is not uncommon for a DBpedia individual to have several redundant properties that need reasoning to be resolved, and reasoning is computationally intensive, especially on a Linked Data counting over 2.3 million individuals. The high dimensionality of large Linked Data also limits their usage for information access purpose, forcing researchers and practitioners in such a field to compute complex rankings of relevant properties and individuals using algorithms, such as *Personalised Page Rank*, on huge networks or with the support of an external text corpus.

Wrapping it up, Linked Data and Ontologies are, at the present state, hard to manage and to well-engineer, they require a lot of effort to be designed, populated, and aligned with other Linked Data, and, on top of that, are still hard to use and sometimes little informative unless heavily processed with statistical techniques.

2.3.2 Vector Spaces, Semantic Similarity and Relatedness

As mentioned in the previous paragraph, distributional information can assist information access systems into making sense of Linked Data, but it can also provide meaningful insights on its own. Vector Space Model (herein VSM) approaches are an alternative to explicit and formal knowledge representations such as the one provided by Linked Data. In VSM entities, instead of being described by a set of predicates, are represented as a vector in a space with a finite number of dimensions. VSM leverage the *distributional hypothesis* of linguistics, which claims that words that occur in similar contexts tend to have similar meanings [75]. Some authors [130] in fact define the meaning of a concept as the set of all propositions including that concept. The VSM was first developed for the SMART information retrieval system [147] and it is now widely used in many different fields. VSMs are commonly used to support several NLP and IR tasks, such as document retrieval, document clustering, document classification, word similarity, word clustering, word sense disambiguation, and many others. The most notable advantage of these techniques over formal representations is that vector spaces can be built in a totally automated and unsupervised way. For a deeper and more exhaustive survey of vector spaces and their usage in state of the art systems, we address the interested reader to [106], [165], and [96].

A notable example of a knowledge intensive task that can be realised with a VSM is the assessment of semantic likeness among entities and concepts. Over the years, the notion of semantic likeness has attracted the interest of the Semantic Web, Natural Language Processing, and Information Retrieval communities [74].

[8] the technical word used by the Semantic Web community to identify grounded terms and conceptual references of real entities.

Two variants have been thoroughly examined: (i) *Semantic Similarity* that can be defined as the likeness of the meaning of two items, for instance, "king" and "president" although not being equivalents have a high semantic similarity since they share the same function, and (ii) *Semantic Relatedness* that can be viewed as a looser variant of semantic similarity since it considers any kind of relationship, as an example, "king" is semantically related to "nation" because a king rules over a nation. Due to the high ambiguity of the very definition of these semantic relationships, it is not unusual to assess relatedness and similarity metrics evaluating their performance on a specific, well-defined and reproducible task [21]. Several metrics can be found in the literature, these metrics are based on statistics [173], set theory [144], and graph theory [138]. One of the most common semantic similarity measures is the *Google Distance* [28] which exploits the popular search engine to compute pairwise similarity between words or sentences. This metric has proven its effectiveness in several knowledge intensive tasks like the evaluation of approximate ontology matching [64]. However, the employment of this metric in real systems is impractical or too expensive due to the extensive usage of the underlying search engine. Additional approaches are based on structured knowledge bases such as taxonomies and ontologies. Wordnet is one of the most exploited tools to compute semantic similarity employing many different methods including machine learning and graph search algorithms. The authors of [21, 22] present an extensive survey of semantic similarity metrics exploiting Wordnet. Many authors propose strategies to compute similarity and relatedness among items exploiting entities included in the LOD cloud. The majority of LOD-based techniques select a limited number of features among the multitude of properties available in the cloud. To accomplish this task, methods such as Personalised Page Rank, are regularly used in the literature [138]. These techniques are often employed in the semantic-based personalisation field, despite being particularly demanding from a computational perspective [121]. Wikipedia has also been widely exploited to evaluate semantic relatedness metrics: in the research paper [63], the authors propose *Explicit Semantic Analysis* (ESA), a technique that employs machine learning algorithms to build vectorial representations of Wikipedia articles through the use of their textual contents. On the other hand, the authors of [174] introduce an alternative to ESA that takes advantage of the links present in Wikipedia articles. Such technique achieves similar performance but requires less data and computational power. In the following Chapter 4 we will present our two metrics to compute the similarity between entities, one for incoming links and one for outgoing ones, the latter one being closely related to the aforementioned Google Distance.

2.4 Towards Information Extraction and Knowledge Representation Localisation

In this chapter we surveyed various techniques to extract information from text and to arrange information into knowledge. However, most of the presented work has been done on the English language and within a Western cultural environment. As already mentioned in Chapter 1, reality is drastically different: the Internet is nowadays used by billions of people from all around the world, speaking different languages and approaching the Web with different cultural backgrounds. In our opinion, extracting relevant information from localised text and building knowledge bases aware of cultural differences is one of the prominent and most exciting challenges of present-day NLP and Artificial Intelligence research.

Aside from the notable example of multilingual KP Extraction case studies already surveyed in Paragraph 2.1.1, there already exist several NLP multilingual resources. One of the best-known of these resources is *WordNet*, a lexical database for the English language [115] that groups English words into sets of synonyms called *synsets*. WordNet has been localised in many different languages including Italian [146] and Arabic [16, 1]. A large number of multilingual Part-Of-Speech tagging (POS tagging) systems are also available like *Brill tagger* [18], *Stanford Log-linear Part-Of-Speech Tagger*[9] [162], *TreeTagger*[10] [150], *CRF-ADF Sequential Tagging Toolkit*[11] [160], and many more. Several stemming libraries can be found too: *Porter's stemmer*[12] [167], *Lovins stemmer* [101], *Lancaster stemmer* (also known as Paice/Husk Stemmer) [133], and *Snowball stemmer*[13] [139] are all examples of available resources. Moreover, several lemmatisation tools have been developed, among others we cite here: *LemmaGen*[14] [82], *MADAMIRA*[15] [135], and *Morph-it-lemmatizer*[16] [66]. Many natural language parser systems have been built as well: *Stanford Parser*[17] [37], *Berkeley Parser*[18] [137], *BLLIP reranking parser*[19] (also known as Charniak-Johnson parser, Charniak parser, Brown reranking parser) [25], and *Egret parser*[20], among all the others. Finally, some research groups and practitioners grouped several tools into a comprehensive suite meant to provide a compact and complete environment to develop NLP applications. Notable examples

[9] `https://nlp.stanford.edu/software/tagger.shtml`
[10] `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`
[11] `http://klcl.pku.edu.cn/member/sunxu/code.htm`
[12] `https://tartarus.org/martin/PorterStemmer/`
[13] `http://snowballstem.org/`
[14] `http://lemmatise.ijs.si/`
[15] `https://camel.abudhabi.nyu.edu/madamira/`
[16] `https://github.com/giodegas/morphit-lemmatizer`
[17] `https://nlp.stanford.edu/software/lex-parser.shtml`
[18] `https://github.com/slavpetrov/berkeleyparser`
[19] `https://github.com/BLLIP/bllip-parser`
[20] `https://code.google.com/archive/p/egret-parser/`

of such suites are: Stanford CoreNLP[21] [107], Apache OpenNLP [22], NLTK (Natural Language Toolkit)[23] [15], GATE (General Architecture for Text Engineering)[24] [33], ScalaNLP [25] [73].

All these works, however, provide only a base layer to build NLP applications rather than ready-made applications. Named Entity Recognition, Named Entity Linking, Word Sense Disambiguation, Ontology Extraction in principle could be all implemented for a large number of languages, assuming that what worked for English will work also in other languages. This is, however, a nontrivial assumption, oblivious of the huge linguistic variability present in the world. The lack of evaluation resources, such as datasets, best practices, and evaluation frameworks hinder the research activity on this topic since it makes hard to assess the actual effectiveness of new techniques.

The aforementioned DBpedia [93], is a large, multilingual, semantic knowledge graph built upon information extracted from Wikipedia in various languages. Despite its size, breadth, and indubitable value for the Artificial Intelligence community, DBpedia is, after all, only a very large set of grounded predicates. In other words, all triples inside it are considered equal, however not all connections among entities, concepts, and events are equal, and their relevance may vary from culture to culture. More importantly, some topics require high cultural sensitivity to be dealt with without, being perceived as offensive or inappropriate by a potentially large part of the Web user base.

With state of the art Semantic Web technology it is impossible to have a culture-sensitive view on a knowledge base without a massive knowledge engineering effort. In fact, relying solely on the technologies offered by the Semantic Web stack introduced in Section 2.3, culture-related information should be first formalised into an ontology, then coded into RDF data, adding new levels of complexity to the already large semantic network of an existing Linked Data, and finally accessed with new, ad-hoc designed, SPARQL queries which will be likely to increase the already high complexity of accessing graph data. All these steps would be nontrivial to implement, thus requiring expert knowledge and vast resources to be accomplished. In our opinion, such a task should be done, instead, automatically, in a more cost efficient way, possibly building on the huge amount of content generated by the Social Web, which is typically localised and heavily influenced by the local culture of the writer. Semantic VSM introduced in Paragraph 2.3.2 are knowledge representation models that can be trained in an unsupervised way upon large volumes of textual data, therefore they look like a promising technique to acquire such a knowledge in an effective and economic way. Using VSM, moreover, implies a fuzzy, non formal way of dealing with knowledge, wherein emerging distributional properties overlook

[21] `https://stanfordnlp.github.io/CoreNLP/`
[22] `https://opennlp.apache.org/`
[23] `http://www.nltk.org/`
[24] `https://gate.ac.uk/`
[25] `http://www.scalanlp.org/`

the logic connections among the considered entities. VSM then seems to be the appropriate tool, given the notoriously complex and delicate matter of representing and describing cultural viewpoints on events, ideals, authorities, and people.

With respect to the challenges highlighted in this chapter, in the next one we will try to address these issues, illustrating a multilingual approach to KP extraction instantiated on Italian and Arabic, and pinpointing the main challenges of evaluating an Information Extraction system on language with a few resources available, leveraging user test and testing over multiple datasets. Subsequently, in Chapter 4 we will introduce a compact and efficient vectorial representation of semantic relatedness, evaluate such a metric over three text corpora, respectively in English, Italian, and Arabic, and assess how well this solution fits the various cultural differences implied by working with an Anglo-American, a Southern European, and a world-wide, but mostly Middle-Eastern user base.

3

Multilingual Keyphrase Extraction

In this chapter, we present DIKpE-G an experimental system specifically built for performing KP Extraction and Inference from textual documents. DIKpE-G can operate on different languages and exploits a knowledge-based approach combining various classes of knowledge, in part language-dependent, in part independent and it is designed to emulate some of the cognitive processes that are exploited when a human expert is asked to summarize or classify a text. The proposed system has been evaluated on the Italian and Arabic language, exploring new evaluation protocols involving users and multiple datasets.

Most of the results presented in this chapter refer to our work published in [49, 47, 42, 77].

3.1 Abstract Keyphrase Extraction Framework

As shown by the limitations presented in Section 2.1, to effectively deal with the peculiar characteristics of the different languages, a preliminary design work of knowledge engineering is necessary. Taking also into consideration our previous work on keyphrase extraction for English texts [140], we developed a *Knowledge-Based* KP extraction technique based upon (i) exploitation of several kinds of knowledge, (ii) consideration of the specific languages addressed, and (iii) typical/common writing styles. The initial design work allowed us to identify four classes of knowledge which can be exploited to recognize meaningful KPs in a text:

1. *Statistical knowledge*: this knowledge deals exclusively with the quantitative aspects of natural languages, such as the frequency of a given word in a text or its inverse document frequency in a corpus; though lacking of a clear semantic meaning, it can be useful to identify terms and phrases that characterize a text.

2. *Linguistic knowledge*: this knowledge comes from the specific language considered and deals with morphological and grammatical aspects of the text; examples of linguistic knowledge are POS tags, the information on whether a given word is a stopword or not, or whether a given sequence of words is con-

stituted by an acceptable pattern of POS tags for a KP (such as, for instance: "noun-noun" or "adjective-noun").

3. *Meta/Structural knowledge*: this knowledge consists of heuristics over the general structure of the text and typically deals with the position of a phrase in the considered document; an example of meta-knowledge is knowing that phrases appearing in the abstract of an article may be more representative than the ones included in its body. This knowledge corresponds to various writing styles exploited by the author of the text. Another example of exploitable meta-knowledge is constituted by some specific metadata inserted in a document by the author (such as the "topic" meta-tag in Web pages and the "subject" meta-tag in a *PDF* file).

4. *Semantic/Social knowledge*: this knowledge comes from sources external to the considered text. Semantic knowledge deals with the meaning of the terms present in the candidate KPs and with the typical conceptual context where they are used. An ideal source of semantic knowledge is constituted by ontologies, which describe concepts, their properties, and their mutual relationships, together with the natural language terminology usually exploited for linguistically referring to them. Other common sources of such kind of knowledge are dictionaries, thesauri, classification schema, etc. This knowledge is useful for recognizing terms belonging to a specific jargon and for resolving polysemic words. Other relevant examples of sources of semantic knowledge, which are becoming more and more popular in the participative Web (Web 2.0), are fast growing collaborative dictionaries, thesauri and knowledge bases, such as DBpedia. They feature a very wide conceptual coverage and they provide a way to socially validate candidate KP: for a candidate KP being an entry of one of these sources, means that other humans have already identified it as a meaningful way to linguistically refer to the underlined concept. This is the reason why we consider appropriate to attach to this kind of knowledge also the term "social".

It is important to point out how such classes of knowledge differ from each other in terms of domain and language dependency: as shown in Figure 3.1 statistical knowledge is both domain and language independent, linguistic knowledge is domain independent, but language dependent, meta/structural knowledge is domain dependent, and, finally semantic/social knowledge may be both domain and language dependent. Domain and language dependency are very different. Domain dependency can be sensibly reduced by considering only general assumptions, such as assuming that most of the interesting concepts of a document will be introduced in its first section. It can also be turned down by taking into account information gathered from dictionaries or ontologies with a very broad scope (such as Wikipedia). Language dependency, on the other hand, cannot be relaxed: language dependent knowledge, indeed, needs dedicated modules and/or knowledge bases.

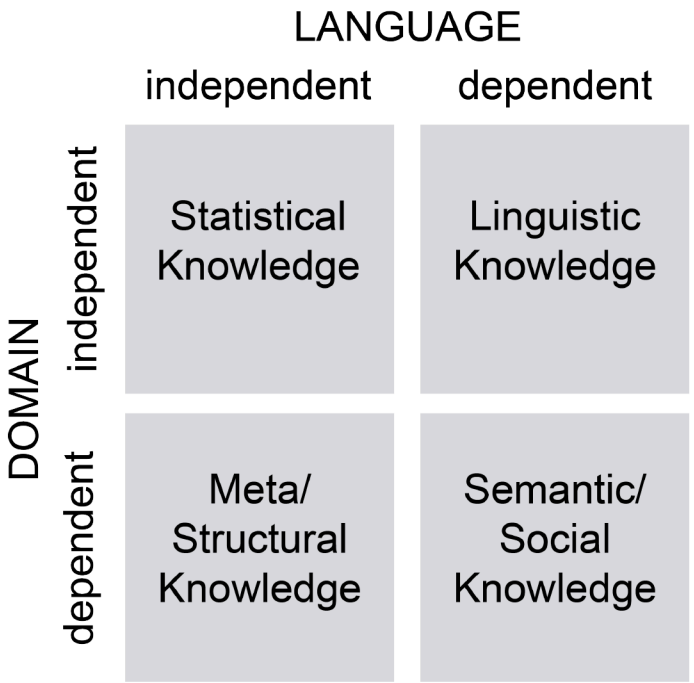

Figure 3.1: Domain and Language dependencies of the various kinds of knowledge considered.

When reading a text with the purpose of extracting relevant concepts a human expert typically performs various kinds of evaluations and we believe that, in order to match the performance of a human, an automatic system should try to follow the same process. To this purpose, the overall KP extraction process is organized into three stages: in the first phase, the text is analysed in order to identify all the possible candidate KPs to be possibly extracted from the text. Later, in a second phase, each candidate KP is scored by associating it to a set of features which are the result of applying the various kinds of knowledge described above to the specific candidate KP. More specifically, each class of knowledge is mapped into one or more features and the final selection criterion of candidate KPs takes into account all the features. The chosen features are then combined to produce a final decision associated to the candidate KP: this can be performed, for instance, by means of a unique score or of a multi-dimensional classification technique. This knowledge-based approach can be used both in a supervised and an unsupervised scenario. In a supervised scenario the feature combination function could be the result of a training activity of a machine learning algorithm (e.g.: Bayesian classifier, Support Vector Machine, Artificial Neural Network, etc.), while in an unsupervised approach it is explicitly known and may be the result of a knowledge engineering activity. Finally, in the third phase other relevant KPs are generated once the major concepts included in the text have been extracted. In this stage, a domain-dependent inference process takes place, able to identify other (usually more general or related) concepts that

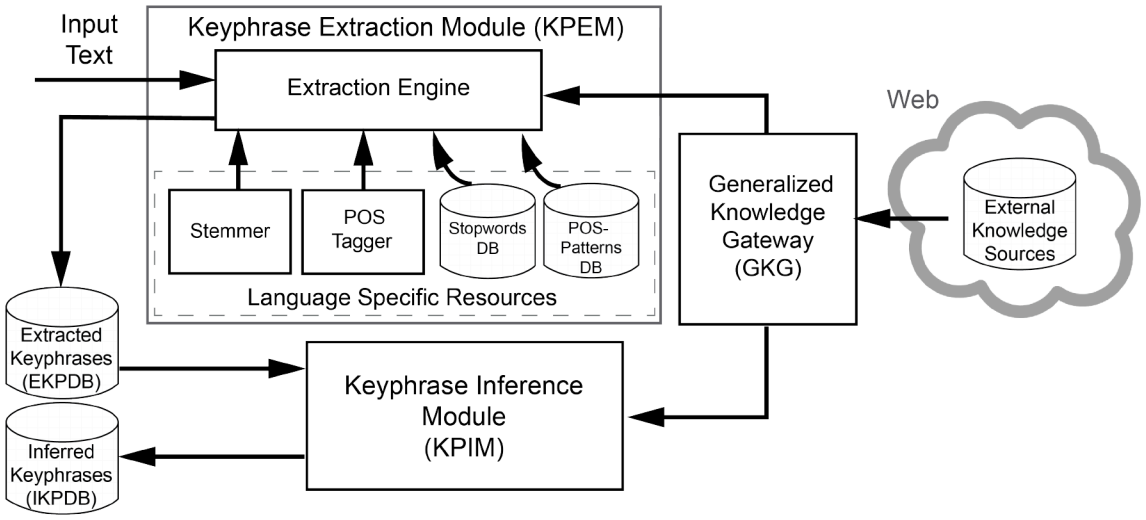

Figure 3.2: Architecture of the DIKpE-G System.

are derived starting from the concepts (KPs) extracted in the first two stages and by exploiting external semantic/social knowledge.

3.2 Multilingual Implementation

The knowledge engineering analyses presented in Section 3.1 allowed us to develop *DIKpE-G* (*Domain Independent Keyphrase Extractor - Generator*) which, as its name suggests, allows the extraction and generation of keyphrases from natural language texts. Figure 3.2 shows the overall organization of the system.

The data workflow mimics the 3-phase cognitive process described in the previous section. First of all the text is read and the *KP Extraction Module* (*KPEM*) discovers and ranks concepts (KPs) that appear in the text, then the *KP Inference Module* (*KPIM*) augments the set of extracted KPs with new linked, related or implied concepts. Operation of DIKpE-G is also supported by *External Knowledge Sources* (*EKS*): in the current implementation we exploit *Wikipedia*[1] and *Wordnik*[2]. The generated KPs represent tacit and explicit knowledge because part of them is explicitly contained in the text and the rest of them are inferred starting from the ones already present in the text.

In order to identify the KPs, the KPEM relies on a series of *Language Specific Resources* (*LSR*). They consist of a *POS tagger* module, a *Stemmer* module and two repositories: one for stopwords and one for POS patterns that typically characterize KPs. Decoupling the language dependent part from the rest of the architecture allows us to easily port the system to other languages. All the necessary language

[1] `www.wikipedia.org`
[2] `www.wordnik.com`

dependent modules are in fact widely available for all major languages: for example, the *Snowball stemmer* library provides functionality for over twenty languages and the *TreeTagger* provides POS tagging for over fifteen languages.

The extraction task is organized in two steps: the candidate KPs selection and the ranking phase. In the first step all possible sequences of one, two, three, and four words are considered, but only the ones matching a valid POS pattern are chosen as candidate KPs. Identification of valid POS patterns is a knowledge engineering task and can be carried out by considering widely used patterns (indicated as "valid") in a large enough set of human generated KPs (human generated such as the author KPs included in scientific papers). The number of POS patterns depends on the considered tag set. Currently, we have a dozen POS patterns for the Italian language, about 40 for the English language, and several hundred for the Arabic language. The difference is due to the different granularity of the employed tag set and the characteristics of the language.

In the following second step, each candidate KP is assessed by means of a set of features, which are computed by exploiting the various classes of knowledge previously described in Section 3.1. In the current implementation of DIKpE-G, we are experimenting with the set of features introduced in [45]. More specifically, in Table 3.1, we show, for the various steps of the extraction, the different classes of knowledge taken into account, the relative features considered and, for each of them, their purposes and value range.

As it can be noticed in Table 3.1, each feature has a value varying in various ranges. Once for each KP a specific set of values have been computed for its features, a final ranking step is performed, which is aimed at producing a final global rank for each KP. The result is a ranked list of KPs: the highest ranked are proposed as relevant keyphrases for the input text. In our vision, the ranking step can be performed in various ways, ranging from (i) a strictly numerical approach to (ii) a more sophisticated and general knowledge-based assessment based on both qualitative and quantitative reasoning. The highly modular architecture of DIKpE-G, allows a seamless substitution of the modules and submodules devoted to ranking, permitting in such a way the experimentation of alternative approaches. The current DIKpE-G prototype follows the approach proposed in [140], which adheres to a numerical approach: each feature is given a numerical value and all the features are then combined in order to compute a unique index called *keyphraseness*, which represents how much a candidate KP is considered suitable and significant for representing the content of the input text. The keyphraseness index is computed in the current DIKpE-G prototype as a weighted linear combination of the features values. The features weights are currently experimentally obtained.

The final phase is devoted to inferring new KPs (i.e. KPs which are not already present in the input text) starting from the topmost ranked extracted KPs. The KPIM considers each extracted KP in order to match it against the entries of the available EKSs: if a match is found (i.e. the considered KP is also an entry of a specific EKS), all the concepts (terms) present in the EKS and linked to the matching

entry are considered as candidate *inferred* KPs. All the candidate inferred KPs collected from all the extracted KPs are then ranked according to the sum of the keyphraseness values of the extracted KPs from which they have been derived. Note that inferred KPs can be obtained both from high-ranked or low-ranked extracted KPs. For instance, the system can infer a KP that is linked to a large number of low-ranked KPs rather than a KP that is linked to a little number of hi-ranked ones. The top-n inferred KPs are finally returned as output together with the extracted KPs identified by the KPEM.

Table 3.1: Usage of the various classes of knowledge proposed in DIKpE-G.

		Knowledge Class	Feature	Purpose	Value Range
KP EXTRACTION	Candidate KP identification	Linguistic Knowledge	POS tag patterns	Excluding certain patterns	
			Stopword list	Excluding certain words	
			Stemming	Working on common stems	
	Candidate KP Scoring	Linguistic Knowldege	POS tag patterns	Preferring typical patterns	0-1
		Statistical Knowledge	Frequency	Preferring most frequent terms	0-1
			Co-occurrence	Preferring common co-occurrent patterns	0-1
		Meta & Structural Knowledge	Phrase depth	Preferring concepts appearing at the beginning of the text	0-1
			Phrase last occurrence	Preferring concepts mentioned at the end of the text	0-1
			Life span	Preferring concepts appearing in a large part of the text	0-1
		Semantic/Social Knowledge	Flag of presence in EKS	Preferring KP appearing in ontologies, dictionaries, thesauri, ...	bool
			Flag of presence in Web 2.0 WKS	Preferring concepts recognized by other human actors	bool
KP INFERENCE		Semantic/Social Knowledge	Navigation paths in EKS	Inferring new KPs related to many extracted KPs	list
			Navigation paths in EKS	Disambiguating polysemic inferred KPs	list

3.3 Evaluation

In this section we describe the various evaluations performed over the presented approach. We are focusing on the evaluation work performed on the Italian and Arabic language. It is important, however, to stress how evaluation of NLP tools for non-English idioms is still a non trivial task, ridden with obstacles posed by the lack of linguistic resources and best practices.

3.3.1 Evaluation Criticalities and Pitfalls

Evaluating an Information Extraction system, in general, is always a delicate and nontrivial task, and KP Extraction in particular has some significant characteristics that hinder its evaluation. The first and foremost difficulty in evaluating automatic KP Extraction lies in the very definition of a KP we used in Chapter 2, which is the one most of the research community refers to. We consider an n-gram a KP when it is relevant within the considered text, but such a notion may very significantly due to subjective factors and pragmatics. To address this issue most datasets are annotated by multiple experts to provide an abstraction layer over the intrinsic subjectivity of the very notion of a Keyphrase. However, involving more and more experts implies raising the cost of building such assets. Several authoritative datasets like Witten99 [175], Frank99 [61], Hulth03 [79], Medelyan06 [113], Nguyen07 [129], Schutz08 [152], Wan08 [171], Marujo11 [109], Marujo12 [108] and more, have been built exploiting this kind of expert knowledge, providing a valuable test ground for the research community. These datasets, however, are all made of English text. Up to now, no authoritative dataset has been established for other languages. As a matter of fact, building new datasets for languages that share only a small fraction of the Web content is not considered a pressing matter and local challenges such as Evalita somehow failed to establish a durable and authoritative benchmark for further research.

Several authors in the literature tried to overcome this obstacle by using the so-called *Wikipedia-based evaluation* [136] which consists in considering Wikipidia surfaces (also known as "link anchors") found inside articles as Keyphrases. While it may sound as a legit approximation, it may introduce severe biases into the evaluation. In fact, Wikipedia surfaces are annotated with the purpose of linking entities, not summarising text content or topics. Consider for instance the case of the English Wikipedia article "England" which has 1329 surface labels[3], while it is true that it is a long article, such an amount of KPs can be safely considered information overload and it is not helpful for any descriptive purpose. On the other hand building an evaluation dataset from scratch is costly and time-consuming.

Recent work in the field of crowdsourcing suggests that a large enough team of workers con build such an artifact with a reasonable cost, however we are referring

[3]All the statistics provided in this section refer to a Wikipedia snapshot taken in September 2015.

to a few seminal works and a lot of research effort is still needed to address the many pitfalls of crowdsourcing the creation of such a resource.

Given this tremendous lack of localised evaluation resources, user test may offer a viable alternative to assess KP extraction quality, however authoritative guidelines to design such tests are missing as well.

In the following of this section we will investigate both user testing and offline testing over multiple non-authoritative datasets.

3.3.2 Italian KP Extraction Evaluation

The Italian language has received little attention from the NLP research community in general and as a direct consequence it lacks both linguistic resources to test new systems on and baseline systems to compare results. At the time of this writing, the main publicly available linguistic resources for the Italian language are: *itWaC* (Italian Web as Corpus) [8] a 2 billion token corpus built by Web crawling and subsequently POS-tagged and lemmatized with automatic tools; the *Repubblica* corpus [7], a very large corpus of Italian newspaper text which consists of 380 million tokens enriched with POS-tagging, lemmatization and categorized in terms of genre and topic; *Paisà* [102] (Piattaforma per l'Apprendimento dell'Italiano Su corpora Annotati), a corpus of authentic contemporary Italian text from the Web made of 250 million tokens fully annotated in *CoNNL* format (lemmatized, POS-tagged, syntactic dependencies); the *CORIS/CODIS* corpus (COrpus di Riferimento dell'Italiano Scritto/COrpus Dinamico dell'Italiano Scritto) [58, 65] which consists of 130 million words from texts in electronic format chosen to represent modern Italian. Many other specific resources exist like EPIC (European Parliament Interpreting Corpus) [117], EUR-Lex [5], and DGT-TM (Directorate-General for Translation - Translation Memory) [158, 159] which are intended for machine translation purposes, Araneum Italicum Maius [11, 12] which includes thousands of texts POS-tagged with Tree-Tagger [151], MultiSemCor [13, 14] which is a collection of English and Italian texts annotated with POS, lemma and word sense that can be used to train machine translation algorithms, POS taggers, Word Sense Disambiguation algorithms, and many more. Unfortunately, none of these assets can be used to test a KP extraction system, since they are meant for completely different purposes.

This severe lack of linguistic resources is probably one of the reasons why meaningful baseline systems cannot be found in the literature. In essence, an offline evaluation of the Italian language is currently meaningless, since it would not be reproducible, due to the lack of shared resources, and it would have no authoritative baseline.

To address the problem of evaluation, we advocate user testing. Building on the insights on evaluation described in the previous section, we defined a novel evaluation protocol. First of all, we decided a task that defined the pragmatics of our KP extraction. In our opinion, this preliminary step is mandatory and should be well-documented because as introduced in the previous section the notion of "relevance"

might change according to the user's goal. Once the task is defined, a meaningful set of documents must be collected. We considered for each document its length, counted in the number of sentences, its topics, selected from a previously defined list of topics of interest, and its writing style, possibly evaluated with a metric such as the *Flesch-Kincaid grade level* [85]. A good set of test documents should provide a realistic distribution of these three parameters. We intend as "realistic" a distribution that reflects the actual characteristics of the targeted domain of application. For instance if we want to test a system for KP extraction in the academic domain we would expect a small variation in the writing style, and a large variability in topics, on the other hand, if we were testing a similar system, but tailored for the sports news domain, a realistic distribution should present a much narrower variability in topics and a much larger one in length and writing style. The evaluation user pool should be chosen carefully as well: it should include people of different age, gender, and cultural background. Again, the distribution of these parameters should match as much as possible the one of the target user segment. For instance, a KP Extraction system for scientific papers should be tested on a user pool mostly made of scholars, while a similar system built for the sports news domain would require a drastically different user pool, reflecting the real consumers of such texts. Finally, user questionnaires must be defined. Developing a meaningful questionnaire is a delicate task: pragmatics, user assumptions, and domain constraints must be taken into account, moreover to the best of our knowledge there not exists best practices, guidelines, and frameworks to build questionnaire for KP extraction assessment. Given these observations, we propose a minimalistic questionnaire wherein the chosen task is clearly described to the test user and for each extracted KP we asked the user if in his opinion it is serviceable to the extent of the described task. If the user deems the KP unsatisfactory we ask him to motive why it is inappropriate, possibly choosing among a selected list of common pitfalls. Despite its simplicity, we deem this kind of questionnaire very informative, since it grants a relevance assessment on each extracted KP, and a justification for the non-relevant ones.

Instantiating this protocol, we decided to tailor the evaluation on the task of document summarisation. Summarisation is one of the most straightforward applications of KP Extraction, and can support significantly the user experience of several information access systems [44]. The test document set included 50 research papers, 11 to 16 KPs were extracted from each document, and the user pool consisted of Master's degree and Ph.D. students. A focus group was organised to pinpoint the most common pitfalls of extracting KPs from scholarly literature for summarisation purposes and five classes of unsatisfactory KPs were identified:

- *Too specific*: the considered KP identifies a concept which is included in the text, but not useful to the extents of summarisation, e.g. a very technical term, a reference to an accessory notion, and so on.

- *Too generic*: the considered KP identifies a very broad or common concept

Table 3.2: Results of user evaluation on Italian KP Extraction.

Evaluation	Frequency
Good	56.28%
Too Generic	14.72%
Too Specific	2.27%
Incomplete	9.85%
Not Relevant	9.85%
Meaningless	7.03%

which is hardly informative in the considered domain, e.g. "system", "research", "experimentation", or "methodology".

- *Incomplete*: the considered KP is clearly a substring of a larger one, e.g. "Adaptive" instead of "Adaptive Personalisation".
- *Meaningless*: the considered KP does not represent a domain concept.
- *Not relevant*: the considered KP does not fall into any of the previously described categories, but still it is not useful to the purpose of summarisation.

This questionnaire was administered to the test users during a testing period of five days, with the results reported in Table 3.2. An example of the questionnaire provided to the users can be found in Appendix B.

3.3.3 Arabic KP Extraction Evaluation

With respect to Italian, the Arabic language has received more attention from the NLP research community, and some baseline systems are available. We decided to test our KP extraction technique on the Arabic language in an offline fashion, exploiting existing systems as a baseline and running the experiments on corpora already known in the literature. All of the existing Arabic KPE approaches, however, have been tested and evaluated against datasets built by their authors. These datasets were built by collecting Web published documents from news portals, wikisites, and scientific articles. After that, KPs are usually assigned to the document collection manually by the authors or by some experts. There is no gold standard dataset for testing and training Arabic KPE systems. We decided not to build a custom dataset to avoid bias. Instead, we used three datasets already known in the literature and described in [53], [122] and [52]. Table 3.3 presents the key characteristics of the considered datasets.

Different experiments have been conducted to assess the performance of the proposed approach with the state of the art. In the first experiment, we benchmarked our system against four approaches. The considered baselines are: KP-Miner, a hybrid method based upon KP-Miner, a distributional approach based on Google's

Table 3.3: Arabic evaluation datasets details

Dataset	Topic	# of docs	Avg. Size in words	Avg. # of KPs
DS1	Leadership and management	27	1227	7.8
DS2	General Wikipedia pages	100	776	7.9
DS3	Agriculture, environment, and food	35	641	11.1

Table 3.4: Comparison between the proposed system and other approaches - Arabic.

	KP-Miner	TF-IDF	Word2Vec	Hybrid	Our System
Avg. Precision	**0.13**±0.06	0.11±0.06	0.09±0.05	0.10±0.05	**0.13**±0.08
Avg. Recall	**0.38**±0.25	0.35±0.24	0.29±0.25	0.31±0.25	0.37±0.25
Avg. Detected Keys	2.49±1.21	2.25±1.16	1.70±0.93	2.00±0.93	**2.53**±1.52

Word2Vec library, and *Term Frequency-Inverse Document Frequency* (TF-IDF) [53, 122]. For each system and dataset, we evaluate precision, recall and the average number of correctly extracted KPs per document; and for all these measures, the mean value and the standard deviation (± SD) are provided. This benchmark was performed over DS2 (see table 3.4).

We can observe how the proposed approach outperforms the baseline ones in terms of correctly extracted KPs per document with an average of 2.53. This result is remarkable considering the characteristics of the DS2 dataset where the number of human annotated KPs per document can vary between 1 and 12.

The second experiment was performed to compare the results of extracting KPs using the lemmatization approach which is employed by our system and the stemming approach which is adopted by Arabic-KEA system [52]. Arabic-KEA is a framework for KPE from Arabic news documents and it is based on the KEA [175]. Since KEA is an open software, it has encouraged many researchers to adapt it to

Table 3.5: Comparison between Arabic-KEA using stemmers and our approach with lemmatizer.

Dataset	Statistical stemmer	Rule based stemmer	Lemmatizer
DS1	1.56±1.59	0.67±10	**2.78**±1.3
DS2	2.58±1.24	1.17±0.94	**3.75**±1.42
DS3	1.4±0.86	0.96±0.87	**2.57**±1.67

Table 3.6: A comparison for the top-5 KPs extracted by TEC and KP-Miner against the proposed approach - Arabic.

TEC Approach[4]			KP-Miner			Our Approach		
KP	Translation	Judge	KP	Translation	Judge	KP	Translation	Judge
موسه تصمه مس موسه تف تگه مع	The right to the freedom	Y	مسمه وگ مد تگ موسه تصمه	Everyone has the right	N	موسه تصمه ه موسه تف تگه مو مه	The rights and freedoms	Y
وگ مد تگ موسه تصمه	one the right	N	موسه مطمه مع موسه تف مو مع	The United Nations	Y	تصمه موسه موس وگ موس	Human Rights	Y
تصمه موسه موس وگ موس	Human Rights	Y	مسمه مو مسه موس	Whereas it is	N	تصمه موسه تصمه موسه مع	Right of protection	Y
مس تگ مو موسه تصمه	one the right	N	موسه تصمه ه موسه تف تگه مو مه	The rights and freedoms	Y	موسه تصمه مس موسه تگه	The right of work	Y
موسه تگ موسه ه تصمه	The universal of rights	N	مسمه مس تگ مو	Everyone has	N	تصمه ه مه وگ موسه مع	Equal rights	Y

other languages.

Arabic-KEA uses two different approaches for stemming: statistical and rule-based stemming. The two systems were run on the three datasets and the average number of detected KPs was computed. The results shown in Table 3.5 suggest that lemmatization consistently produces more correct KPs than stemming.

Finally, the quality of the top-5 extracted KPs was compared against those detected by KP-Miner and TEC approach [4] using the Arabic version of the Universal Declaration of Human Rights (UDHR) which was used by TEC author. Table 3.6 shows the result of the comparison. A native Arabic speaker judged the quality of each extracted KP stating whether the KP could be consider accettable (Y) or not (N). TEC and KP-Miner detected only 2 good KPs out of five while all of the five KPs extracted by our system are good. For TEC, the reason of extracting bad KPs is that it did not consider the syntax feature of the sentences. For example, the second and forth KPs have two words which exist in two different NPs but their frequency of occurrence in the document is high. Also, KP-Miner depends mainly on frequencies and uses customized stemming, so a lot of extracted KPs contain stopword.

3.4 Final Remarks

In this chapter we presented a multilingual approach towards Keyphrase extraction and evaluated the developed testbed systems over two languages, Italian and Arabic, that up to now received little attention from the NLP community. This work highlighted two critical issues of KP extraction on non-English idioms, namely the lack of a framework that can compactly describe KP extraction and ease the development of new approaches, and the lack of evaluation best practices.

3.4.1 Towards a Multilingual Framework

The abstract framework described in 3.1 has been instantiated onto the Italian and Arabic language, providing a clear separation between linguistic, statistic, and heuristic considerations involved in the KP Extraction process. From an engineering point of view, this abstraction allowed us to better design and implement the described systems, maximising code reuse. As a matter of fact a number of statis-

tic and heuristic insights can be safely considered invariant with respect to the language and therefore the software modules encoding them should be built to be highly reusable.

Building on these insights, and considering the massive localisation effort that the NLP community need to tackle in the future due to the rapid growth of the non-English content, we strongly advocate the establishment of a shared multilingual NLP framework. In the case of Information Extraction tasks, such as Keyphrase Extraction, a framework to ease the engineering and the development of localised applications would be extremely valuable, given the large number of end-applications that can benefit from Information Extraction, i.e. recommender systems, search engines, tutoring systems, and many more. Moreover, since the vast majority of software engineers around the world are not NLP experts as well, providing some degree of abstraction over established tools and techniques, like the ones introduced in the previous chapter in Section 2.4, thus making them a commodity, would be extremely valuable to foster the localisation of Information Extraction and Natural Language Process in general.

Unfortunately, available state-of-the-art systems tend to provide a "one-size-fits-all" solution, that is either a very vertical solution tailored for one target application or it is a generally domain independent application that does not allow domain-specific business logic to be introduced. To the best of our knowledge, none of the currently available solutions can be easily tailored to fit new languages. Even most of the applications regarded as frameworks are very vertical and far from being friendly for those who do not have an extensive NLP background: for instance LingPipe[4] offers a comprehensive set of Machine Learning algorithms commonly used for IE tasks, however its lack of abstraction over the techniques to be used makes it extremely hard to integrate into other applications, on the other hand, the Stanford NLP pipeline is a monolithic application and can be used as a tool rather than a framework. The authors of [30] propose *CURATOR*, an NLP framework that allows to annotate text in various ways. However such a framework is far from being an integrated solution and its primary focus is to organise low-level text processing tasks such as sentence splitting, tokenisation, and POS tagging, lacking support for higher-level tasks such as KP Extraction. Implementing the insights described in this chapter into a concrete framework would, therefore, be useful to the NLP and IE community.

3.4.2 Definition and Evaluation

Evaluating KP extraction where no authoritative and established datasets are available is indubitably a nontrivial task, ridden with pitfalls that may introduce significant bias in the observed outcome and therefore intrinsically prone to be disputed.

The first pitfall we encountered lies in the very definition of what a Keyphrase

[4] `http://alias-i.com/lingpipe/`

is: a string of words that capture the main topic of the document. What should be intended as the main topic of a document is utterly dependant on the user's task, cultural background, and ability to understand and interpret a text.

This fact is evident even in the authoritative data provided by the SEMEVAL challenges 2010 and 2017: in the 2010 dataset a set of scientific papers was annotated with its topics carefully chosen by both their authors and readers [84], while in the 2017 a set of paragraphs was annotated with the tasks, methods, and materials therein described or mentioned [3]. As a matter of fact, these two datasets represent different tasks that can be both considered as KP extraction but, ultimately, cannot be compared or merged to train a system.

Pragmatics, therefore should be emphasised, because it appears to be a fundamental component of the very definition of what a Keyphrase is. Unfortunately most datasets, therein included the aforementioned SEMEVAL datasets, lack a clear indication of the task pragmatics and it has to be guessed by either reading multiple times the accompanying paper or by looking at the data themselves.

The second pitfall we encountered is characteristic of non-English languages and consists in the already mentioned severe lack of evaluation resources, may they be datasets or guidelines. Of all the multilingual works cited in Chapter 2 none appears to have been evaluated on a previously known and established dataset, with most authors tailoring a new dataset specifically for their evaluation. While such a practice seems inevitable when no previous resources are available, it is also prone to cherry picking[5], and can hardly be defended no matter how many baseline systems can be considered. User testing, on the other hand, could be a viable solution to assess KP extraction quality, but again the lack of established best practices and evaluation frameworks seriously hinders the significance and reproducibility of the experiments.

In this work we explored solutions to overcome both evaluation criticalities: we evaluated our Arabic KP extraction system over different datasets, providing evidence that our approach consistently offers satisfactory performance, and we thoroughly documented the user evaluation design we implemented allowing other researchers and practitioners to adopt it in the future.

[5]The practice of providing incomplete evidence

4

Referential Space Models

In this chapter we are introducing our approach to semantic distances. Our technique leverages the hypothesis that concepts that are frequently mentioned within the same documents tend to be tightly related. We call this assumption the *Reference Hypothesis*. Such an assumption can be used to build, from a large enough text corpus, a semantic vectorial space we will refer to as *Referential Space*, however, this model tends to have an extremely high dimensionality, thus some optimisation is needed. We tested our approach on multiple languages and evaluated its ability in identifying and ranking related concepts to popular, well-known entities. The crowdsourced evaluation highlighted how our approach seems to provide on average better results than LOD-based ones over the three considered languages. Another distinctive feature of this approach is its ability of embedding in a compact vectorial representation the perception of distances between entities and concepts as seen by the authors of the texts included in the considered training corpus. Ideally, training different vectorial spaces upon different text corpora written by different authors, coming from a different cultural background, but dealing with the same topics should provide us with different, culturally sensitive, views of the said topics.

In this chapter we are introducing Referential Spaces, describe a dimensionality reduction technique to reduce the computation of semantic relatedness between two entities in the referential space to constant time, provide an evaluation of the perceived quality of our approach, and assess its ability of representing the different perceptions of distances and relationships between topics occurring in different cultures.

The work presented in this chapter has partly already been published in [48, 46] and it is partly unpublished.

4.1 Referential Spaces

As shown in chapter 2, most literature work on word spaces leverages the distributional hypothesis, that is that words occurring in similar contexts may yield similar meaning. However to overcome their limitations and to exploit the potential of hypertextual connections we introduce a new hypothesis: the *Reference Hypothesis*. We assume that entities that are referenced in a similar set of documents might

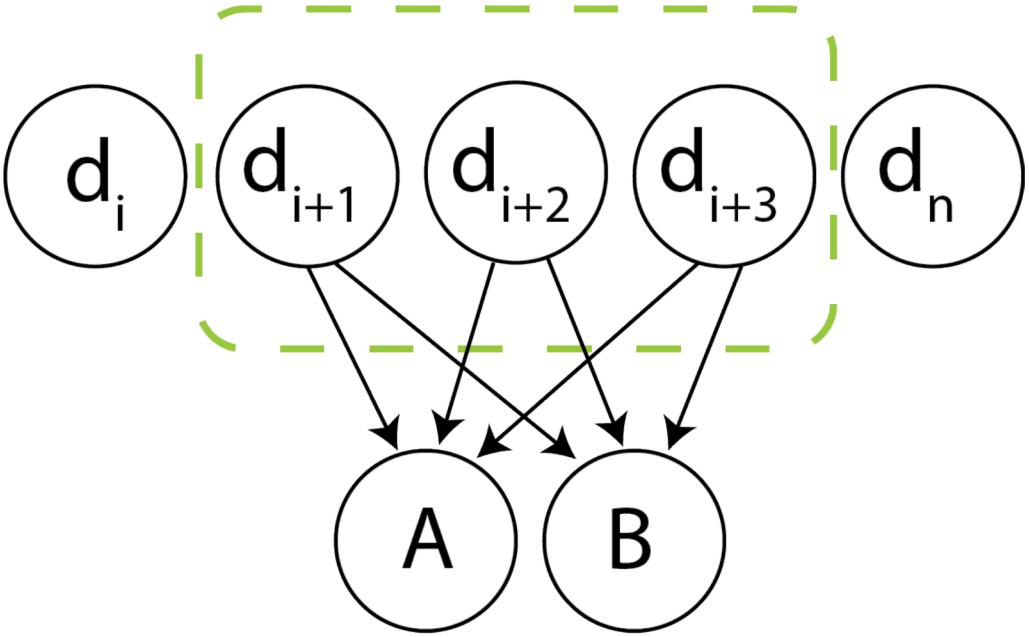

Figure 4.1: Two entities referenced by the same set of documents.

yield strong semantic affinity. For instance, in Figure 4.1 two entities (A and B) are referenced in the same documents: this implies a semantic affinity between A and B.

This assumption is motivated by the fact that intuitively referencing something in a document implies the referenced item to be relevant in the context of the document, therefore entities that get constantly referenced together are relevant in the same contexts, hence they might be semantically related. This hypothesis can be seen as a generalised version of the aforementioned distributional hypothesis, however we would like to stress how even though words can be seen as entities, entities can be intended as way more abstract items, for instance other documents or ontology entries. For instance, the reference hypothesis applies to the scientific literature since articles citing similar sources are very likely to deal with similar topics. Other works in literature embrace this assumption though not formalising it, such as [80] wherein a scientific paper recommender system exploiting co-citation networks is presented.

Building a vector space exploiting the Reference Hypothesis is straightforward once a large enough corpus of documents annotated with hyperlinks is provided. Within the corpus two sets must be identified: the *entity set* E and the *document set* D; the first includes all the referenced entities, while the latter the considered annotated documents. The vector space is represented with an $E \times D$ matrix that initially is a zero matrix. Iteratively, for each $d \in D$ all the references to elements in E are considered, and for each $e \in E$ referenced in d, the (e, d) cell of the matrix is set to 1. Since referencing a given entity only once in a document is a typical best

practice in several domains[1] we are not considering how many times e is referenced in d. Once all documents are processed we obtain a matrix where each row represents all the references to a given entity: we call such matrix *Reference Matrix* and the vector space it generates *referential space*.

It is important to point out that, as long as the considered corpus is made of HTML pages, there is no need of annotating the texts. Hyperlinks can be conveniently parsed without performing NLP tasks such as tokenization, stemming, linguistic analysis, and so on. Furthermore, corpora of cross-referenced hypertext documents where documents form a network of connections exist on the Web. Some of these corpora are particularly interesting to analyse under the Reference Hypothesis, because $E \equiv D$ since any entity is also a document, resulting in a square, although not symmetrical, matrix. On the other hand, if the considered corpus is not annotated with hyperlinks, there exist technologies such as *TagMe* or *Babelfy* that allow automatic annotation with links to ontology entries. In this scenario, however, heavy NLP is involved and for a very large corpus this solution might be impractical. Another relevant feature of hypertextual connections is that they are provided with a *surface* label, that is a word or a string of words to be clicked to open the linked page. The surface label represents the natural language label associated to the linked entity and typically this is a many-to-many relationship: an entity can be referred with different surface labels as well as a surface label can link to different entities in different contexts. Entities represent the meaning of surface labels, while surface labels represent the signifier of entities. We call the multiplicity of meanings of a surface label its *ambiguity*.

Evaluating the similarity of two entities in such a vector space reduces to computing the distance between their vectors. Countless distance metrics exist in the literature such as norms, cosine similarity, Hamming distance, and many others surveyed in [172]. All these metrics can be used in the Reference Matrix, however we prefer the Jaccard similarity coefficient (also known as Jaccard index [81]), defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{4.1}$$

where A and B are sets of items. Since each entity $e_i \in E$ can be considered a binary vector, it can also be expressed as the set that contains all the document $d_j \in D$ such that $(e_i, d_j) = 1$ in the Reference Matrix. The similarity of two equal sets is one, whereas the similarity between two sets that have no elements in common is zero. The choice of the Jaccard similarity coefficient is motivated by the intrinsic simplicity of such a metric and by the evidence presented in the literature that the Jaccard index performs better than other methods for finding word similarities in VSM approaches [92, 106].

[1]For instance in Wikipedia only the first time an entity is referenced it is annotated with a hyperlink, and in literature bibliographies have no duplicate entries.

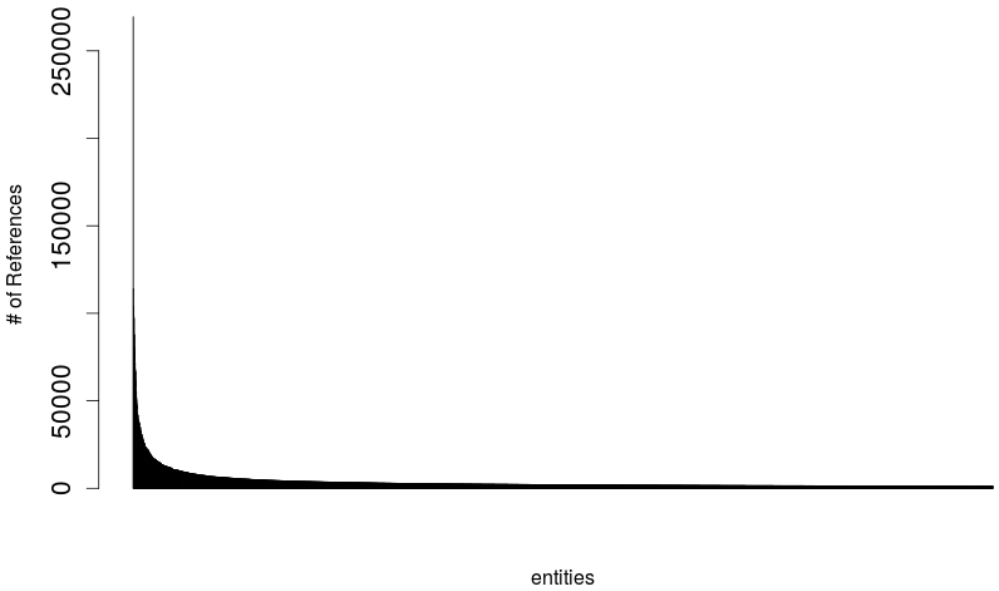

Figure 4.2: Distribution of page references in the 5000 most referenced English Wikipedia pages.

4.2 Dimensionality Reduction

Computing the Jaccard index is linear in the size of the considered vectors, which can be extremely large when considering big corporas such as Wikipedia.

Wikipedia is the largest human annotated cross-referenced[2] text corpus that can be practically downloaded and which is freely available on the Internet. This allows an extremely relevant case study due to both the good properties of the corpus and its size: the English Wikipedia alone consists of over 8 millions articles and in the rest of this section we will refer to it to illustrate the issues related with processing Wikipedia-size corpora. All Wikipedia articles are considered to identify the document set that, being Wikipedia cross-referenced, equals to the entity set as well; revision pages and other documents that have no encyclopedic value are not considered. The vector space is then constructed as illustrated in the previous section by parsing all articles and the final result is a square matrix, wherein each article is associated with a set of other articles referencing it.

The dimensionality of such a matrix is over 8 millions, which is the count of English Wikipedia's encyclopedic articles[3]. This Reference Matrix is also highly

[2]Here we intend as "reference" any hyperlink present on the page, not limiting to the homonymous section commonly included in Wikipedia articles.

[3]All the statistics provided in this section refer to a Wikipedia snapshot taken in September 2015.

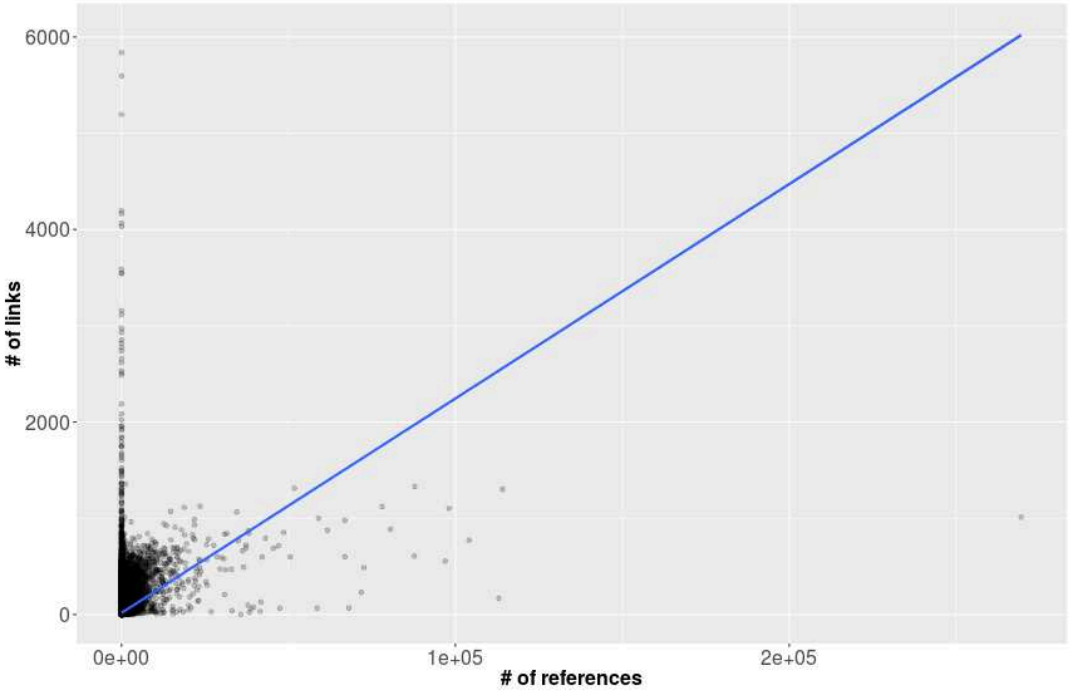

Figure 4.3: Distribution of page references and links in the page.

sparse, with few entities being frequently referenced (with a peak of over 269000 references for the article "*United States*") and the vast majority getting only a handful of references: while the average number of references to an entity is 9.77, the median is only two, and the 75% of the considered entities have at most four references. This fact is illustrated in Figure 4.2 where the distribution of references to the 5000 most referenced Wikipedia pages is shown and the power law like trend is evident. There is also a loose correlation between how many links are included in a page and how many times that page is referenced, indicating that frequently referred entities often correspond to articles that point to many other entities. This situation is pictured in Figure 4.3 where it is shown that most pages have few links and are seldom referenced as well, while only a small set of articles holds the majority of connections. Given these distributions, it could be tempting to assume the computation of the Jaccard index in the average case to be very efficient, since the average number of items in a set of references is less than 10 and very large sets are uncommon. This is, however, not a sound assumption. Let us assume Wikipedia's language usage being representative of the real-world language usage, which is reasonable due to its collaborative nature. Entities with a large number of references are intuitively the most used ones, therefore they are the most likely to be found in a text, mentioned in a conversation, or used as a search query. Given this observation, we can safely assume that in a realistic application scenario most similarity checks would be between entities with very large reference vectors. The

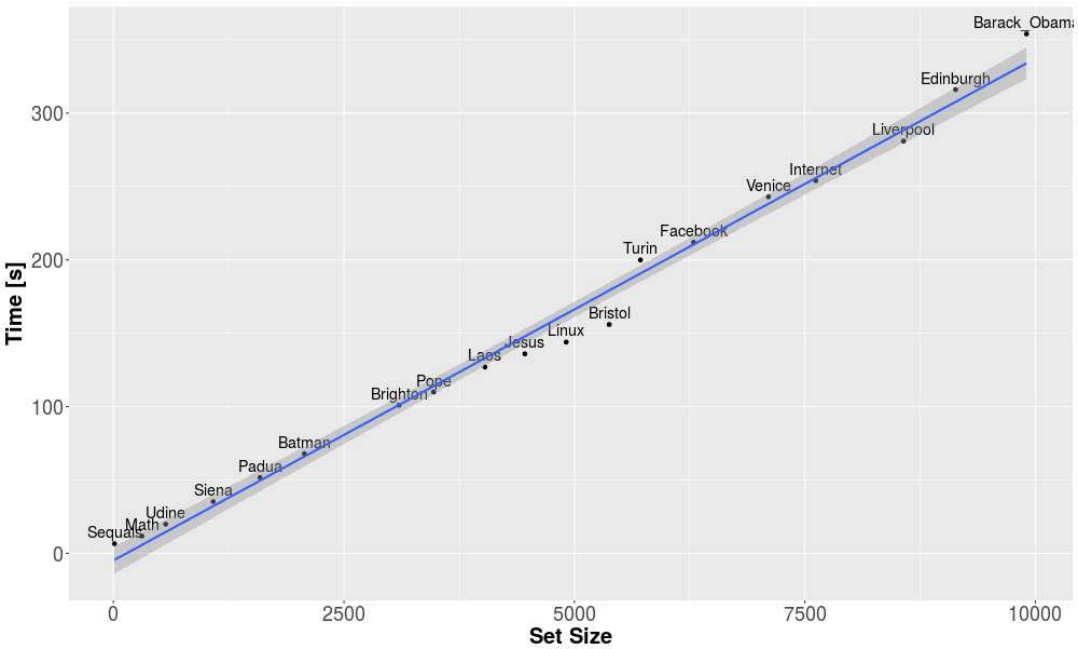

Figure 4.4: Time taken to compute the Jaccard index of a sample of Wikipedia pages with increasing vector size over all other Wikipedia pages (Correlation = 0.995).

linearity of the Jaccard index, therefore, cannot be ignored for real-world usage. To better illustrate the criticalities introduced by such a dimensionality, let us consider for instance the task of computing a ranking of the most related entities to a given one. In Wikipedia, performing such a task implies computing millions of set unions and intersections given the very large number of entities therein included. Given the linearity of the Jaccard Index computation, we expect the time needed to perform such a task to grow in a linear way with respect to the size of the considered item. This is clearly visible in Figure 4.4 where the computation times of computing a rank of the most related entities is shown for several query entities with a growing reference set size. Obviously, taking over 300 seconds to compute a list of related items for *Barack Obama* is not feasible in a everyday usage scenario, especially considering that Google takes milliseconds to perform the same task.

The computation of the Jaccard index can be reduced to constant time using the MinHash optimisation [19]. Such a technique allows to efficiently compute the similarity between sets without explicitly computing their intersection and union. Its most common form consists in using a hash function to map each element of the set to an integer number and then selecting the minimum as a representative of the whole set. The probability that two different sets share the same minimum with respect to the hash function tends to the Jaccard similarity coefficient between the two sets [95]. The more hash functions are used, the closer the estimate gets to the real Jaccard similarity coefficient value. In this work, we used 256 distinct hash functions to achieve a fine enough approximation of the Jaccard similarity coefficient.

This translates to representing each entity as a 256 positions vector. Such a vector can be considered as an entity's fingerprint in the considered text corpus and implies a significant dimensionality reduction with respect to the initial vectorial space which may count millions of dimensions. This optimisation allows our method to scale up as the number of considered entities grows: being the number of positions of the fingerprint vector constant, checking semantic similarity between two entities will take constant time. With respect to other solutions presented in the literature such as [174] wherein the evaluation of semantic similarity is polynomial with respect to the size of the considered knowledge base, the MinHash optimisation significantly reduces the complexity of such an operation. As a matter of fact, checking which items are the closest ones to a given entity implies checking the target entity against all items present in the knowledge base. With our solution this operation is linear with respect to the knowledge base's size, with other solutions it is quadratic in the best case.

4.3 Perceived Quality Evaluation

Similarly to [21], we evaluated the perceived quality of our system upon a specific application, in this case the retrieval of a set of neighbour entities for exploratory search purposes. Our evaluation activity, due to the intrinsic subjectivity of the very concept of semantic relatedness, was user-based. Two experiments are presented: in the first one, we asked users to give an overall ranking to a list of related items, while in the second one we asked users to assess the relatedness of each item in a given list to a target entity. Such an evaluation was performed over two datasets with different characteristic features and with two substantially different user groups to test the effectiveness of our methodology in different situations, thus preventing data overfitting and cultural biases in the presented conclusions.

4.3.1 Experimental Design

Three hyperlinked text corpora were considered: the English Wikipedia, the Italian Wikipedia, and the Arabic Wikipedia. The English Wikipedia is a well known and massive collaborative encyclopedia, counting over 8 million articles contributed by users from all around the world. On the other hand, the Italian and Arabic ones are curated by users that mostly reside in their local territories and they are a substantially smaller corpus, counting respectively around 2.2 million and 0.8 million articles. We considered these three datasets because they differ significantly in size, in language, and in the user base that generated them.

Using the technique described in Section 4.2, a testbed system, herein named Referential Space Model (RSM), was developed and trained on Wikipedia, associating to each of its items a representative vector. Building on the results of [174] that provides evidence of the importance of both incoming and outgoing links, we

also developed an alternative model relaxing the distributional hypothesis and considering outgoing links, i.e. the items mentioned in the article corresponding to a given item. We refer to this second testbed system as *RSM.outnode*. We chose as baseline two of the most popular search engines on the market[4]: *Google* and *Bing*. One of the most prominent features of said search engines is, in fact, the ability to leverage the LOD cloud to improve search results, more specifically they can retrieve a neighborhood of items closely related to the search query given by the user. To obtain fair and generic search results i.e. not influenced by the recorded browsing history, preferences, and location, Google and Bing search process was depersonalized to prevent the search engines from customizing the final result. Unfortunately, these systems do not cover all languages, in particular, Bing doesn't offer this service for the Arabic language. Therefore in the Arabic experiments, we only use Google's related search as a baseline.

To assess the quality of our two alternative approaches we constructed a dataset of the most visited Wikipedia pages. As a reliable source of data we used the list of Wikipedia Popular Pages[5] that maintains a set of the most accessed 5000 pages on the English Wikipedia and it is updated weekly. For the Italian and Arabic language we retrieved the data from *Topviews Analysis*[6] that keeps track the most visited Wikipedia pages on a daily and monthly bases. For our dataset we focused, for English, Italian, and Arabic on the most *stable items* during the year 2015. We define the stable items as the Wikipedia pages that constantly appear in every weekly/daily version of that list throughout the year, and so receiving constant interest from the visitors of Wikipedia. A set of 1583 stable items were identified for the English language, a set of 4361 items for the Italian, and a set of 2648 for the Arabic. Six evaluation datasets, two for English, two for Italian, and two for Arabic were built by randomly selecting from each language's stable items list 100 items (used in experiment 1) and 25 items (used in experiment 2) upon which all of the systems are able to retrieve related items.

4.3.2 Overall Relevance Assessment

The goal of our first experiment was to assess which one of the systems under investigation produces the overall best set of related items given one search key. To this extent, we considered datasets of 100 items. The crowdsourcing experiment was designed as follows: for each of the considered items, a page was generated including the name of the item, a brief description, a picture, and a box including the results produced by the different systems i.e. four lists of five semantically related items (three lists in the case of Arabic). We decided to show only five results for two reasons: firstly both Bing and Google show at least five related items, which means that for some search queries no more than five items will be

[4]Ranking provided by Alexa: `http://www.alexa.com/`

[5]`https://en.wikipedia.org/wiki/User:West.andrew.g/Popular_pages`

[6]`https://tools.wmflabs.org/topviews`

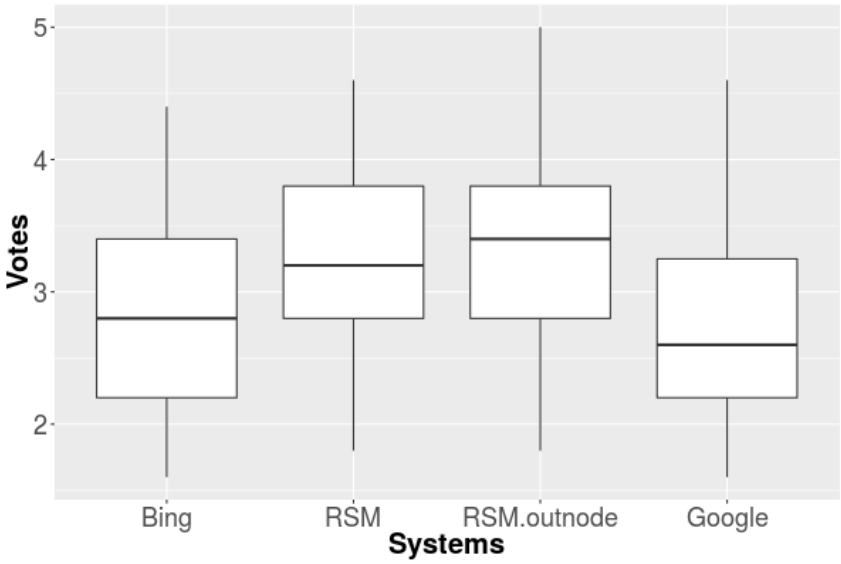

Figure 4.5: Distribution of worker's judgement for the overall relevance assessment experiment - English.

shown, secondly it is a known fact that users typically pay attention only to the top spots of search results lists, with the top five items attracting most of the attention[7]. To avoid cognitive bias, the names of the systems were not shown and the presentation order was randomized, so that the worker had no means of identifying the source of the presented item lists and couldn't be biased by personal preference or previous evaluations. The workers were then asked to rate the item lists according to their perceived quality in terms of relatedness on a discrete scale from 1 to 5 where 1 meant total randomness and 5 that all presented items where perceived as strongly related. Each one of the 100 items in the dataset was shown with the same related items lists to 5 distinct users and their judgements were averaged per system to mitigate subjectivity of judgement. The experiment was performed using the popular crowdsourcing platform *Crowdflower*[8] and iterated on the English, Italian, and Arabic datasets. In the English iteration 32 users from 18 different countries were involved, with an average of 15.62 judgements per user. In the Italian iteration, instead, were involved 59 users from 8 countries, with an average of 8.47 judgements per user. Finally, in the Arabic iteration, 64 users from 14 countries were involved, with an average of 7.81 judgements per user. The distribution of the worker's judgement is shown in Figure 4.5 for the English language, in Figure 4.6 for the Italian language, and in Figure 4.7 for the Arabic language.

[7] `https://chitika.com/google-positioning-value`
[8] `http://www.crowdflower.com/`

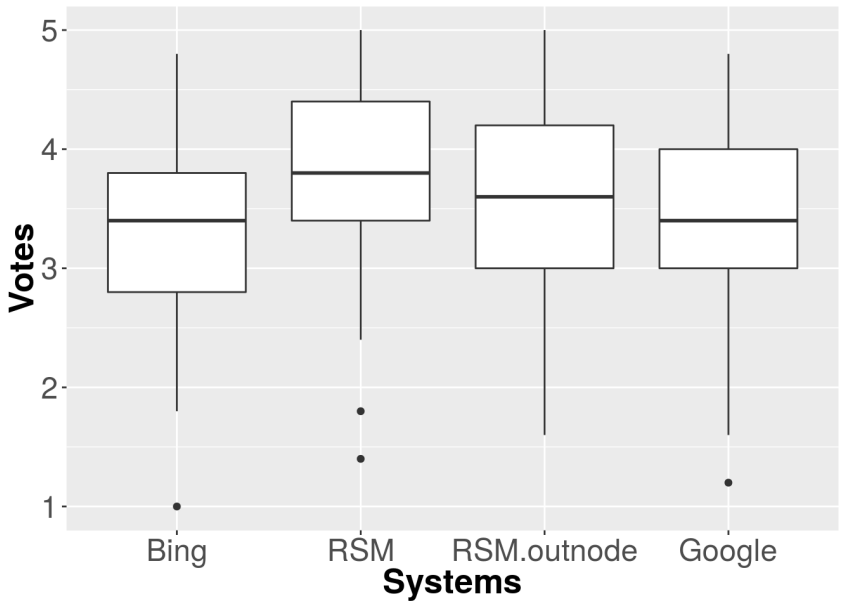

Figure 4.6: Distribution of worker's judgement for the overall relevance assessment experiment - Italian.

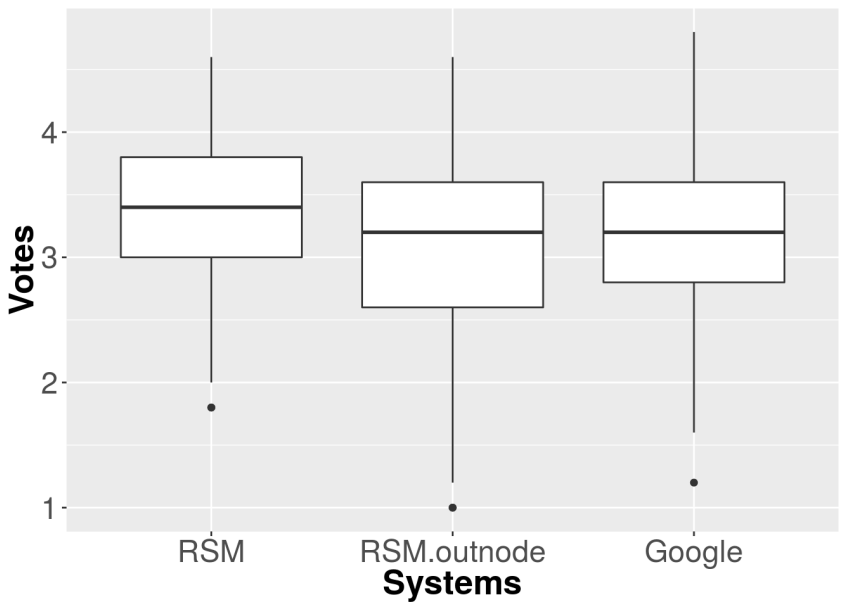

Figure 4.7: Distribution of worker's judgement for the overall relevance assessment experiment - Arabic.

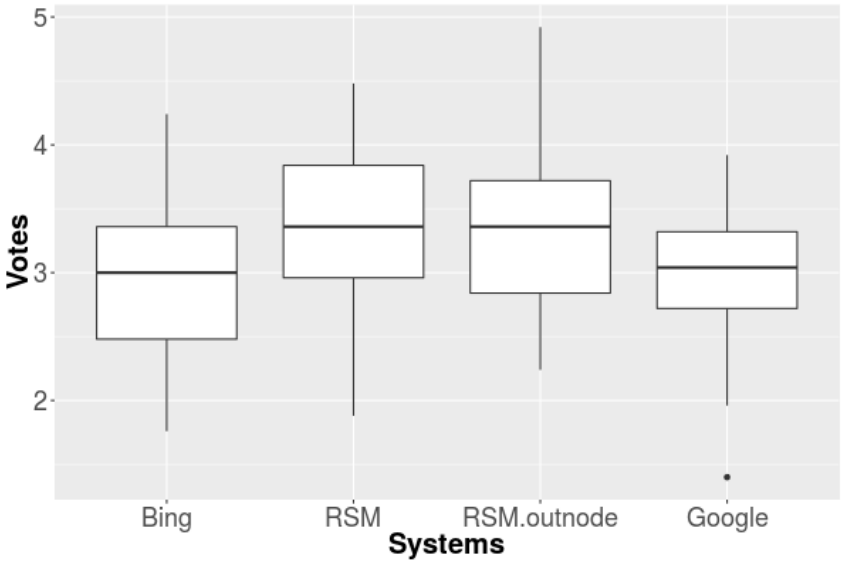

Figure 4.8: Distribution of worker's judgement for the item by item relevance assessment - English.

4.3.3 Item by Item Relevance Assessment

The goal of our second experiment was to assess the perceived quality of each item included in the related items list. To this extent, we considered datasets of 25 items. The experimental setup was similar to the previous experiment, using the same platform and displaying the same information about the target entity (i.e. title, description, and picture). Instead of a list for each system, this time the workers were shown a single list generated by one system only and were asked to rate each item in the list on a scale from 1 to 5 where 1 implied complete unrelatedness and 5 a very high perceived relatedness. The name of the system that generated the list was not shown to avoid bias. A hundred related items lists, 75 in the case of Arabic due to lack of Arabic support in Bing, where therefore generated and human-rated item by item. Again, each item was judged by five distinct users to mitigate subjectivity of judgement. This second experiment was again iterated three times (on English, Italian, and Arabic) and involved by design substantially more workers to further abstract over subjective experience and thus obtain a more impartial judgement. In the end, 146 workers from 38 countries were involved with an average of 3.42 judgements per user in the English experiment, 109 workers from 14 countries with an average of 4.59 judgements in the Italian one, and 97 workers from 18 countries with an average of 5.15 judgements per user in the Arabic experiment. Figure 4.8 shows the distribution of workers' judgements for the English experiment, Figure 4.9 shows the distribution for the Italian experiment, and Figure 4.10 shows the distribution for the Arabic one.

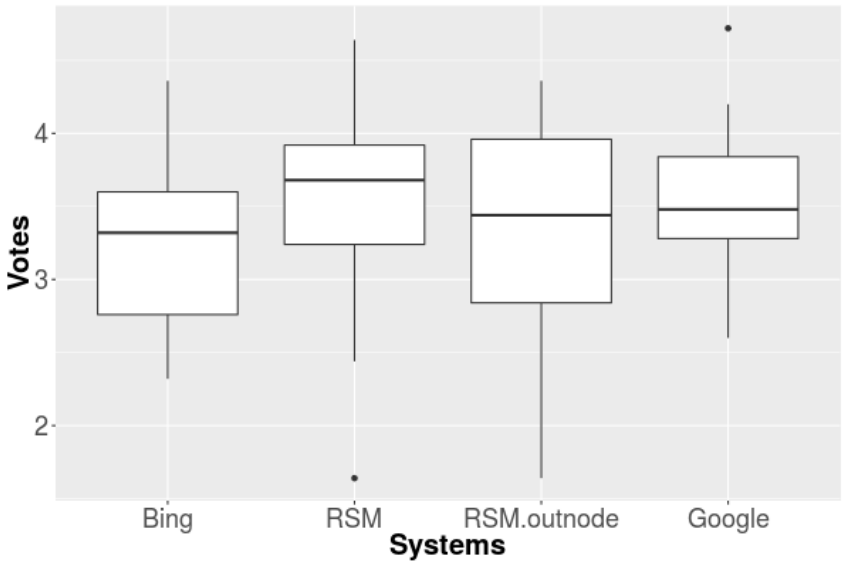

Figure 4.9: Distribution of worker's judgement for the item by item relevance assessment - Italian.

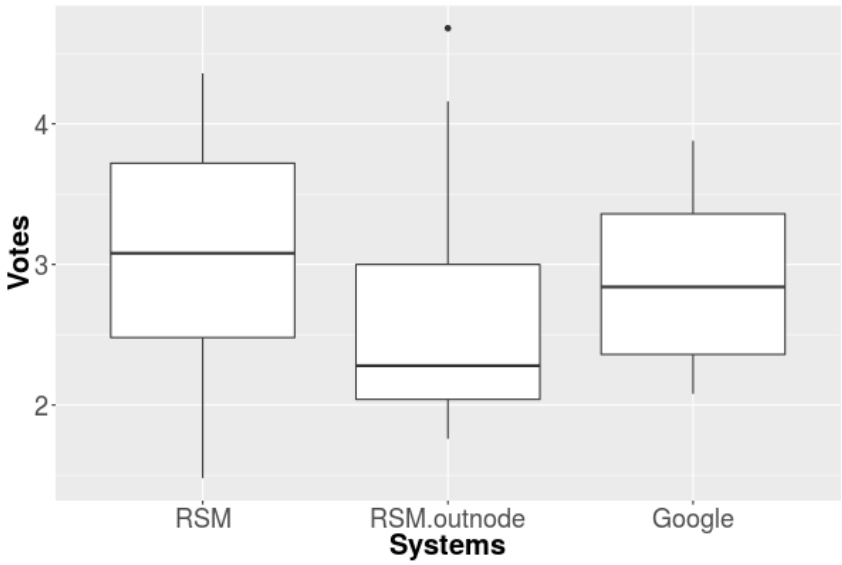

Figure 4.10: Distribution of worker's judgement for the item by item relevance assessment - Arabic.

Table 4.1: Statistical significance of the difference between the considered systems over the English corpus. The upper half of the matrix shows the p-values, the lower the p-values with the Benjamini & Hochberg correction.

	RSM	RSM.outnode	Google	Bing
RSM		0.1896	<0.0001	0.0001
RSM.outnode	0.2275		<0.0001	<0.0001
Google	<0.0001	<0.0001		0.6838
Bing	0.0003	<0.0001	0.6838	

4.3.4 Discussion

The data gathered with the experiments described in Section 4.3.2 and Section 4.3.3 provide some interesting insights on the effectiveness of the proposed technique.

Overall list quality

The results of experiment one showed how our testbed systems RSM and RSM.outnode can achieve satisfactory performance in the considered scenarios. In the English part of the experiment RSM and RSM.outnode achieved, on a scale from 1 to 5, respectively a 3.20 and 3.33 average perceived quality, while Google and Bing respectively 2.79 and 2.82. The statistical significance of the judgement distributions shown in Figure 4.5 was evaluated as well showing how while there is a substantial difference between the perceived quality of our systems and the baseline ones (Bing and Google), between RSM and RSM.outnode there is no statistically significant difference. More specifically the Welch Two Sample t-test was used and produced the results shown in Table 4.1, where in the upper right half of the matrix are shown the p-values produced by the test, and in the lower left half the same values recomputed with the Benjamini & Hochberg correction for multiple hypothesis testing [10]. According to these results, Google's and Bing's related items lists are perceived almost as identical in terms of quality, while our testbed systems' outputs receive a better appreciation by the crowdsourced workers. Moreover, while RSM.outnode appears to achieve a higher perceived quality than RSM on average, the statistical significance analysis shows that such a difference is unlikely to be significant in the current experimental setting. In terms of overall perceived quality the neighbourhoods of related items to a given search key produced by RSM and RSM.outnode do not differ significantly in terms of perceived quality, but there is evidence that consistently outperform the benchmark systems offered by Google and Bing.

In the Italian part of the experiment a similar outcome was observed, with two notable differences: expressed scores were substantially higher for all systems and in particular results produced by Google received a generally more favourable reception with respect to the English part of the experiment. While the former outcome may be ascribed to cultural factors, since the whole judgement distribution is skewed

Table 4.2: Statistical significance of the difference between the considered systems over the Italian corpus. The upper half of the matrix shows the p-values, the lower the p-values with the Benjamini & Hochberg correction.

	RSM	RSM.outnode	Google	Bing
RSM		0.0079	0.0013	<0.0001
RSM.outnode	0.0158		0.6835	0.0141
Google	0.0039	0.6835		0.0308
Bing	<0.0001	0.0125	0.0369	

Table 4.3: Statistical significance of the difference between the considered systems over the Arabic corpus. The upper half of the matrix shows the p-values, the lower the p-values with the Benjamini & Hochberg correction.

	RSM	RSM.outnode	Google
RSM		0.0061	0.1609
RSM.outnode	0.0184		0.1751
Google	0.1751	0.1751	

towards higher scores, the latter suggests that the localised versions of Google and Bing may differ in the used data or retrieval technique. As a matter of fact, the English Bing and Google received very similar judgements, see Table 4.1, and the provenance of the related items lists was unknown to workers to avoid confirmation bias, thus the significant difference observed in the Italian experiment, shown in Table 4.2, implies substantial differences between the English and the Italian versions of the two search engines. On the other hand, the RSM model appears to be the one producing the best received related items lists, while RSM.outnode and Google present no statistically significant difference. The statistically significant difference between the perceived quality of the lists generated by RSM and RSM.outnode in this setting can be ascribed to a substantial reduction in the size of the training data. Overall, RSM is perceived as the best system, RSM.outnode and Google are on par, and Bing is perceived as the worst one.

Regarding the Arabic experiment, the results shown in Figure 4.7 suggest that RSM performs slightly better than RSM.outnode and Google. However, as shown in Table 4.3, the difference between RSM's and Google's rating distribution appears to be not statistically significant. Therefore we can state that RSM outperforms RSM.outnode and is perceived roughly equivalent to Google.

Information Gain Analysis

The results of experiment two support the evidence provided by the previous one. In the English part of the experiment, items retrieved by RSM and RSM.outnode on average score a 3.41 out of 5 on perceived quality while Bing and Google stop at 2.93

out of 5. In the Italian part of the experiment, instead, items retrieved by RSM score an average of 3.6 out of 5, RSM.outnode and Google are tied around 3.5, and Bing scores around 3.4 on average. Finally, in the Arabic leg of the experiment, items retrieved by RSM score an average of 3.1 out of 5, RSM.outnode 2.2, and Google 2.8. These numbers, however, provide little information being average values of perceived quality of item ranked in different positions. Looking at the whole distribution of judgements shown in Figure 4.8, 4.9, and 4.10, the high variance of the various distributions can be easily noticed. Such a variance can be justified by the fact that all items included in the generated lists are considered and rated. However, not all positions of a result list are equal to the extents of exploratory search. To address this issue we evaluated the Normalized Discounted Cumulative Gain (NDCG) of the considered systems. NDCG is a metric commonly used in IR to assess a search engine's performance basing on the comparison between an ideal list of the most relevant retrievable items and the actual list produced by the evaluated system. Its core idea is that the higher the position of an item in the result list the more important the quality of that item should be in the quality evaluation of the system, therefore the presence of scarcely relevant items in the top spots tends to "punish" the evaluated system. The ideal list was computed by considering, for both parts of the experiment, for each of the 25 search keys, all the items retrieved by the four systems, picking the five ones that on average received the highest user ratings and ordering them in descending average rating order. The distribution of the NDCG values scored by the four considered systems over the search queries included in the datasets is shown in Figure 4.11, Figure 4.12, and Figure 4.13 and its detailed statistics are presented in Table 4.4, Table 4.5, and Table 4.6. These results support the evidence brought by the first experiment as well. In the English part of the experiment, RSM and RSM.outnode provide consistently results perceived as more relevant than the ones brought by Google's and Bing's tools. Again, there is no statistically significant difference in the average perceived quality between RSM and RSM.outnode (p-value = 0.68) and between Google and Bing as well (p-value = 0.88). On the other hand, the statistical significance between RSM and Google, RSM and Bing, RSM.outnode and Google, and RSM.outnode and Bing is high with p-values below 0.0001. The NDCG analysis shows how, despite scoring being on average on par with its RSM.outnode counterpart, the RSM system has the smallest variance in the perceived relevance of its results, implying that it is less likely to produce results perceived as poor on a single-try basis. In the Italian part of the experiment, instead, RSM achieves substantially higher nDCG scores than its RSM.outnode counterpart, which, again, presents a vary large nDCG score distribution and, on average, performs slightly worse than Google's related items search, though its median nDCG value is higher than Google's. Like in the previous experiment, the RSM model appears to be able to cope better with changes in training data. Finally, also the Arabic data confirm the results of the previous experiment, with RSM.outnode achieving significantly worse performance than RSM and Google, with the former slightly outperforming the latter with an average NDCG

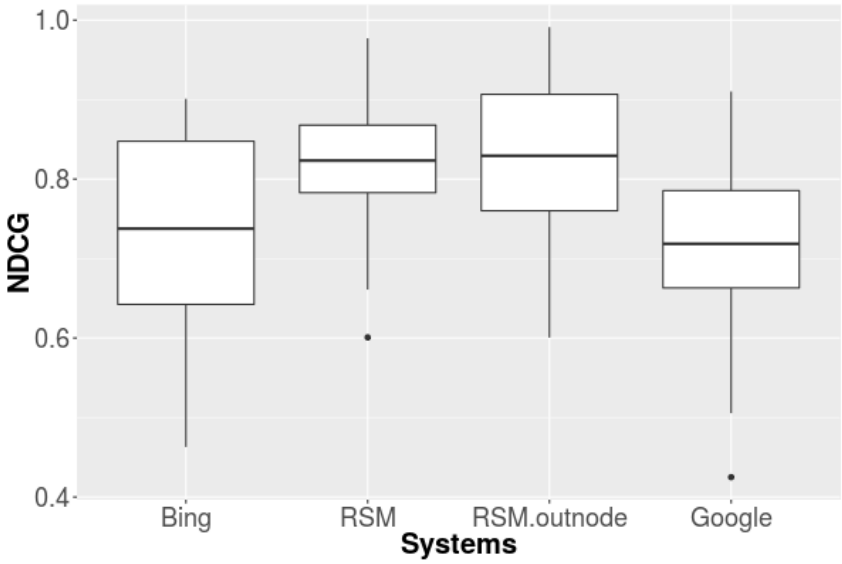

Figure 4.11: NDCG values distribution evaluated on the results of the item by item relevance assessment experiment - English.

of 0.85 against the average 0.82 scored by Google.

The statistical significance analysis showed in Table 4.7 (English), Table 4.8 (Italian), Table 4.9 (Arabic), confirms as well the insides gather from the item by item relevance assessment experiment discussed in the previous paragraph.

Finally, it is important to stress how the MinHash optimisation allowed us to move the complexity of a pairwise similarity measurement from linear to constant. This means that without the said optimisation it would be computationally demanding to retrieve items semantically related with a lot of connections. Consider for instance the Wikipedia article about Barack Obama which, at the time this article being written, contained over 250 links and was referenced over 9900 times by other Wikipedia articles: without MinHash it takes over 300 seconds on our test machine[9] to generate a list of semantically related items, while with the MinHash optimisation it takes less than a second on the same machine. Moreover, the constant complexity of MinHash allows it to seamlessly scale up to larger knowledge bases. While our approach allows this optimisation to be made retaining quality results, other metrics, such as the ones presented in [174, 63], do not.

4.4 Culture Sensitivity Evaluation

The evaluation work presented in the previous section provides evidence of the perceived quality of the entity neighbourhoods that can be identified using the proposed knowledge representation. The gathered insights allow us to state that in general

[9] An Intel I7 with eight cores and 32GB RAM

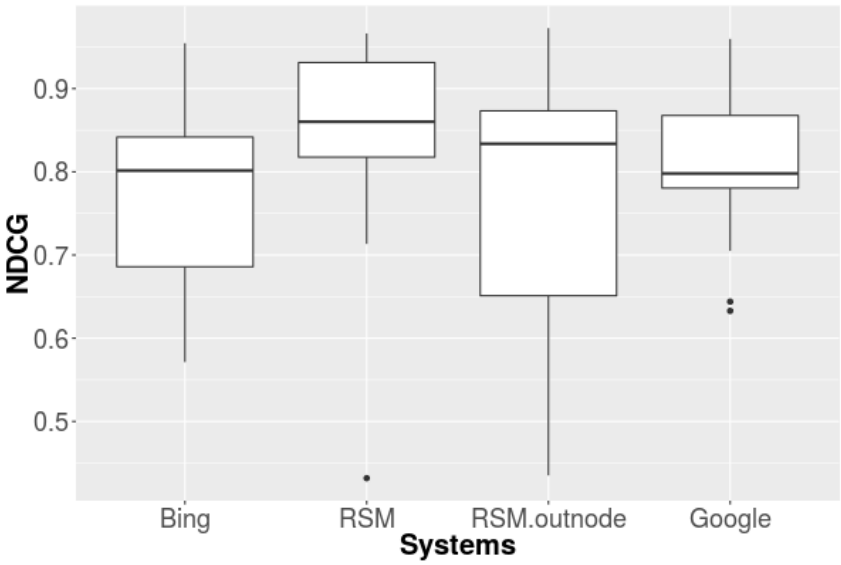

Figure 4.12: NDCG values distribution evaluated on the results of the item by item Relevance assessment experiment - Italian.

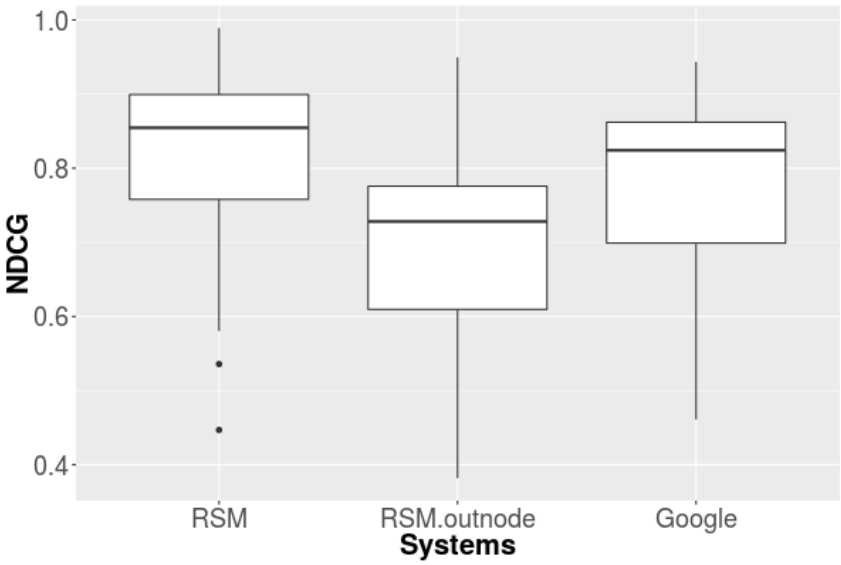

Figure 4.13: NDCG values distribution evaluated on the results of the item by item relevance assessment experiment - Arabic.

Table 4.4: Distribution statistics on NDCG evaluation - English.

	Bing	RSM	RSM.outnode	Google
Minimum	0.4629	0.6009	0.6006	0.4250
1st Quartile	0.6423	0.7829	0.7601	0.6631
Median	0.7376	0.8232	0.8293	0.7186
Mean	0.7247	0.8113	0.8226	0.7196
3rd Quartile	0.8475	0.8678	0.9066	0.7855
Maximum	0.9010	0.9771	0.9910	0.9102

Table 4.5: Distribution statistics on NDCG evaluation - Italian.

	Bing	RSM	RSM.outnode	Google
Minimum	0.5714	0.4319	0.4352	0.6329
1st Quartile	0.6859	0.8177	0.6511	0.7804
Median	0.8015	0.8602	0.8338	0.7980
Mean	0.7793	0.8493	0.7664	0.8121
3rd Quartile	0.8418	0.9313	0.8733	0.8677
Maximum	0.9546	0.9664	0.9726	0.9598

Table 4.6: Distribution statistics on NDCG evaluation - Arabic.

	RSM	RSM.outnode	Google
Minimum	0.4469	0.3819	0.4611
1st Quartile	0.7581	0.6096	0.6990
Median	0.8546	0.7282	0.8242
Mean	0.8167	0.6954	0.7751
3rd Quartile	0.8994	0.7758	0.8621
Maximum	0.9889	0.9496	0.9435

Table 4.7: Statistical significance of the difference between the considered systems over the English corpus in the item by item relevance assessment experiment. The upper half of the matrix shows the p-values, the lower the p-values with the Benjamini & Hochberg correction.

	RSM	RSM.outnode	Google	Bing
RSM		0.9653	0.0073	0.0083
RSM.outnode	0.9855		0.0128	0.0141
Google	0.0212	0.0212		0.9855
Bing	0.0212	0.0212	0.9855	

Table 4.8: Statistical significance of the difference between the considered systems over the Italian corpus in the item by item relevance assessment experiment. The upper half of the matrix shows the p-values, the lower the p-values with the Benjamini & Hochberg correction.

	RSM	RSM.outnode	Google	Bing
RSM		0.1896	0.7769	0.0781
RSM.outnode	0.3339		0.2226	0.7923
Google	0.7923	0.3339		0.0761
Bing	0.2344	0.7923	0.2344	

Table 4.9: Statistical significance of the difference between the considered systems over the Arabic corpus in the item by item relevance assessment experiment. The upper half of the matrix shows the p-values, the lower the p-values with the Benjamini & Hochberg correction.

	RSM	RSM.outnode	Google
RSM		0.0409	0.2994
RSM.outnode	0.1228		0.1757
Google	0.2994	0.2636	

the sets of related entities retrieved with our Referential Space model are preferred to the ones retrieved by Google and Bing leveraging their knowledge bases. The distances between entities, as described in Section 4.1, are evaluated upon distributional consideration on their usage in a document corpus. Distances between entities, therefore reflect the actual usage of terms in the considered texts and quantify the relatedness expressed in natural language. Ideally, considering a body of text somehow representative of a culture should allow us to build a representation of the distances among concepts perceived by that culture. Considering multiple corporas where the same entities, or at least similar entities can be found, should allow the construction of multiple versions of the Referential Space model specifically intended for users of the different cultures considered.

To verify these assumptions, we ran some additional evaluation, comparing the results provided by the various Relational Spaces built to run the experiments described in Section 4.3, estimating their overlap. Given the results of the aforementioned evaluation, we can already assume the related concepts retrieved by our approach to be considered relevant by users of the language examined. Therefore we are evaluating here their diversity rather than quality. To assess the diversity of the retrieved related entities and concepts, we considered 50 entries of Wikipedia that can be deemed as culturally sensitive topics, such as “Religion”, “Terrorism”, “Women”, and “Freedom”. Aside from these rather general topics, the list included notorious and controversial historical characters and events as well. For each entry of the list, a set containing the ten most related entities was extracted from each

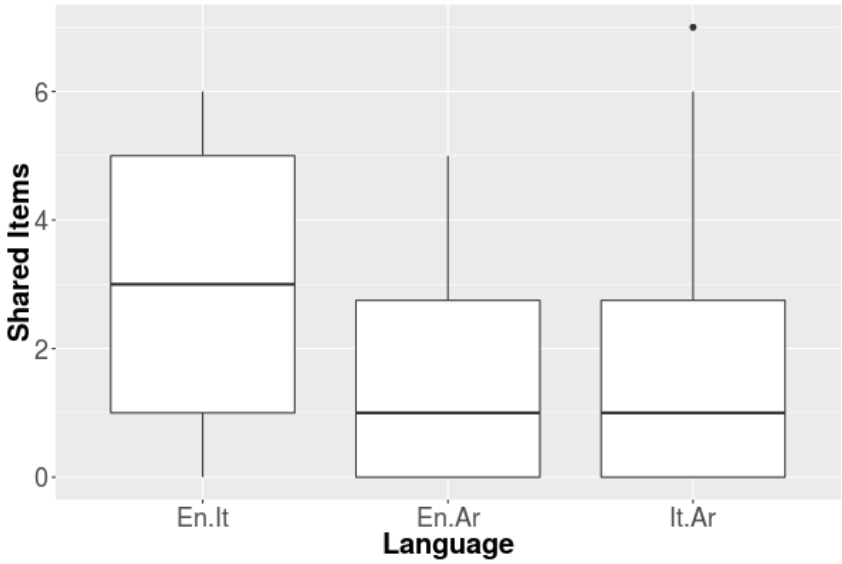

Figure 4.14: Similarity between related items set retrieved by the three considered models.

Referential Space model considered, i.e. the ones built from English Wikipedia, Italian Wikipedia, and Arabic Wikipedia. We defined as diversity metric between two sets the average number of overlapping concepts: the lower, the more diverse we consider the sets. We consider as overlapping concepts the ones that are equivalent, i.e. that can be safely translated one into each other by navigating the cross-lingual Wikipedia link. In Figure 4.14 and Table 4.10 the results of this evaluation are shown. It can be easily noticed how English and Italian models tend to identify slightly more overlapping concepts, on average 2.84, while the Arabic model tends to present more distinctive items, sharing on average 1.54 concepts with its English counterpart, and 1.66 with the Italian one. It is also interesting to note how in the observed sample there were cases (43 instances out of 150) where there was no observed overlap between the results produced by the different models and that no perfectly overlapping result sets were observed. These results provide evidence that different localisation of our VSM present substantial differences, implying that perceived relationships among concepts vary from culture to culture.

To better illustrate how changing the text corpus affects the semantic relatedness evaluation, introducing culture sensitivity, we present a notable example where no overlapping concepts can be found. Table 4.11 shows the ten most related concepts to "terrorism" according to the Referential Spaces built on top of English Wikipedia, Italian Wikipedia, and Arabic Wikipedia. It can be easily noticed how the English set focus mostly on topics dealing with people and organisations involved in the September 11 attacks and their aftermath, the Italian set on the so-called "Years of Lead", i.e. the period of social and political turmoil in Italy that lasted from the late 1960s until the early 1980s, marked by episodes of political terrorism, finally the

Table 4.10: Distribution of shared items on different languages - RSM.

	En-It	En-Ar	It-AR
Minimum	0.00	0.00	0.00
1st Quartile	1.00	0.00	0.00
Median	3.00	1.00	1.00
Mean	2.84	1.54	1.66
3rd Quartile	5.00	2.75	2.75
Maximum	6.00	5.00	7.00

Arabic set is mostly focused on the Yemenite civil war and on people connected to events of the Arab Spring, with the notable inclusion of the Charlie Hebdo Shooting.

Another interesting fact that emerges from a closer observation of the results produced by the three systems is the different type of entities mostly associated with a topic. A notable example can be found in the results returned by the system trained on the Arabic Wikipedia when asked to retrieve semantically related items to "Muslim Brotherhood": among the top ten items can be found several involved people such as "Hassan al-Banna", the movement's founder, and "Yusuf al-Qaradawi" who had a prominent role within the movement's intellectual leadership and hosts a very popular programme broadcast on Al-Jazeera followed by 60 million Arabic speakers worldwide. On the contrary, the English and Italian systems associate the same topic with the events of the Arab Spring and the political leaders overthrown by the uprising, reflecting an outsider perspective on the topic. Similarly, Italian and Anglo-American political movements are associated by the model trained on their language with prominent figures, similar movements, and national events, reflecting an insider perspective. In general, one can easily notice how local public figures and events are associated with other local public figures and events in the localised knowledge model, while in the models trained on other languages the same items are associated with international events and more generic concepts. Similar considerations apply also to popular culture phenomena such as cinema, music, and television broadcast where each model reflects the common wisdom of the user base that contributed to various versions of Wikipedia.

These comparisons allow us to state that there exist substantial differences among the various models we obtained by training our model on different localisations of Wikipedia. However, we wanted to assess the differences among the considered localisations also for the benchmark systems, namely Bing's and Google's "People also search for" box. To measure the result set diversity, we adopted the average number of overlapping concepts, as done for our system. We ran our evaluation on the same set of 50 culturally sensitive topics used before to be able to compare the results. Unfortunately, as pointed out in the previous section, Bing does not offer related item search for the Arabic language, thus we limited its evaluation to English and Italian. Bing showed an average number of overlapping concepts of 8.09

Table 4.11: Results returned by the RSM system for the topic "Terrorism"

En	It	Ar
Al-Qaeda	Red Brigades	Charlie Hebdo shooting
State terrorism	Operation Gladio	Ahmad Awad bin Mubarak
Osama bin Laden	Islamic fundamentalism	Al-Qaeda in the Arabian Peninsula
Civil liberties	Life imprisonment	War on Terror
Counter-terrorism	September 11 attacks	Khaled Bahah
Visa (document)	Years of Lead (Italy)	Mohammed Ali al-Houthi
Money laundering	Cosa nostra	Military
Hezbollah	Mossad	Abdullah II of Jordan
The Pentagon	AK-47	General People's Congress Party (YE)
Taliban	Cold War	Abdul-Malik Badreddin al-Houthi

Table 4.12: Distribution of shared items on different languages - Bing.

	En-It
Minimum	6.00
1st Quartile	7.00
Median	8.00
Mean	8.09
3rd Quartile	9.50
Maximum	10.00

between English and Italian, way higher than the 2.84 reached by our technique. The distribution of the observed overlap is shown in Table 4.12 and it can be noticed that perfectly overlapping sets were observed and that the minimum overlap observed is 6 shared items.

On the other hand, Google shows little or no culture sensibility at all over the three considered languages: with one notable exception, Google provides always the same set of items, translating their names. The only observed exception consists of the results provided by googling "Quaran": while in Italian and Arabic the first item provided is "Bible", in the English version such an item is absent from the list, while all the others are the same provided for the other languages considered, shifted by one position. Table 4.13 shows the observed distribution, which is very close to a uniform distribution, suggesting that Google is the least culture sensitive tool considered.

4.5 Final Remarks

In this chapter we introduced Referential Space models and tested their ability to produce culture sensitive and satisfactory results in terms of assessing semantic

Table 4.13: Distribution of shared items on different languages - Google.

	En-It	En-Ar	It-AR
Minimum	10.00	9.00	10.00
1st Quartile	10.00	10.00	10.00
Median	10.00	10.00	10.00
Mean	10.00	9.92	10.00
3rd Quartile	10.00	10.00	10.00
Maximum	10.00	10.00	10.00

relatedness between entities and concepts. Referential Space models are a distributional representation of knowledge and, as introduced in Chapter 2, they provide a fuzzy representation of knowledge, wherein the proximity between two items is determined by the distribution of references rather than the outcome of a reasoning process. This is generally perceived as a drawback of distributional information since, on the other hand, formal knowledge representation allows to justify the outcomes of a reasoning task. However, when speaking of commonsense knowledge, like the one included in Wikipedia, this is, in our opinion a major advantage since there is no easy way to formalise it and, in general, deal with it in a formal way. As shown in the previous section, in Table 4.11, different models trained on different Wikipedia instances return drastically different results for the same item, despite the fact that most of the possible related items appear on all the three localisations of Wikipedia (in the case of Table 4.11 all the returned items can be found in all the considered versions of Wikipedia). No matter the culture, "Al Queda" and "Red Brigades" are both terrorist organisations, therefore, from a logical point of view they should share the same relatedness with the concepts of terrorism, however it is highly unlikely for an American user to associate immediately "terrorism" to the "Italian Red Brigades", while an Italian user would be surprised of not finding them in a list of topics closely related to terrorism.

The distributional approach captures exactly this kind of differences between cultures and cultural backgrounds, providing a valuable knowledge base for systems that seek a high level of localisation.

In our vision, this kind of knowledge representation is not an alternative to the formal one provided by more formal techniques, but rather a complement. Let us consider once more the case of Wikipedia: aside from being available in several languages, Wikipedia has also a formal counterpart, the famous DBpedia [93]. In DBpedia, every Wikipedia entry is represented as an OWL individual with several properties that provide a formal representation of the textual information presented in its relative article.

It could be therefore tempting to decorate such individuals with additional information, namely its coordinates in various referential spaces. This would allow to represent multiple, culture sensitive, topologies of the DBpedia knowledge graph

allowing culture sensitive topological visits. Such a solution would combine the advantages of both the representations and allow to scale the knowledge embedded in DBpedia, which is multilingual by design, upon different cultures as well. A similar operation could be performed on different knowledge bases as well once large enough annotated text corpora are identified.

5
Conclusions

In this thesis work we described:

- An abstract framework to describe the conceptual blocks of Keyphrase and Key-entity extraction.
- Multilingual Keyphrase extraction techniques built on top of the aforementioned abstract framework.
- The referential hypothesis and its implications.
- An efficient implementation of a Referential Space Model.

With respect to the challenges introduced in Chapter 1, we can state that our expectations were met.

In fact, the abstract KP Extraction framework described in Chapter 3 allows abstraction over several aspects including syntax, morphology, and writing style. This abstraction is provided by the separation introduced between the various kinds of knowledge employed in the Information Extraction process; statistic, linguistic, meta/structural, and semantic/social. Arranging the text processing activity according to our framework and maintaining the separation between modules exploiting the aforementioned types of knowledge allows the construction of horizontal applications that can be robust to style nuances and require a minimal effort to be localised into other languages.

As far at it regards commonsense knowledge differences, adopting the Referential Space models described in Chapter 4 allows to build a knowledge base by processing a corpus of hypertextual documents. When a large enough corpus can be deemed representative of a community's background knowledge, a Referential Space model can be trained in an unsupervised way, providing a compact, yet powerful, representation of that community's culture. When multiple knowledge representations are available, each one of them portraying the perception of a specific group of people, a user can be addressed towards the one he should perceive as more familiar, achieving culture sensitivity.

Finally, the heavy computational times implied by the usage of Semantic Web technologies are avoided by adopting a distributional knowledge representation that can be optimised with local sensitivity hashing techniques such as MinHash.

Wrapping it up, the NLP techniques described in this thesis have the characteristic of providing an abstraction layer over both the language and the culture. This does not mean that these aspects are ignored, rather they are encapsulated within a single component that can be easily integrated into more complex applications. This is a particularly desirable feature when tackling complex and cross-cultural problems, such as information access, matchmaking, or cyberbullying detection to name a few.

5.1 Future Work

In the following of this chapter, as future developments and applications of the work presented in this thesis, we introduce an implementation of the abstract framework described in Chapter 3 and some examples of possible usages of the Referential Space models introduced in Chapter 4.

5.1.1 Implementing the framework

The abstract framework introduced in Chapter 3 is based upon the idea that there exist multiple and diverse kinds of knowledge. This principle can be implemented by encapsulating each knowledge-driven text transformation into an object called Annotator. Annotators can be treated as building blocks to build complex chains and therefore achieve complex IE tasks, as presented in [43] and [9] where the *Distiller* framework is presented. Distiller is an high-level IE framework, built upon the insights presented in this work that can handle several high-level IE tasks.

Distiller is implemented in Java, since such language is widespread among the research community and offers reasonable performance and multiplatform support. Moreover, since it runs on the JVM, Distiller can be used with other popular languages such as Groovy, Scala, and Ruby[1]. Distiller relies on the Spring framework to handle dependency injection allowing easy Web deployment on Servlet containers such as Apache Tomcat.

Distiller is organized in a series of single-knowledge oriented modules, where any module is designed to perform a single task efficiently, e.g. POS tagging, statistical analysis, knowledge inference, and so on. This allows a highly modular design with the possibility of implementing different pipelines (i.e. sequences of modules) for different tasks. All these modules are required to insert the knowledge they extract on a shared blackboard so that a module can use the knowledge produced by another module. For example, an n-gram generator module can generate n-grams according to the POS tags produced by a POS tagger module. Since these modules work by *annotating* the text on the blackboard with new information, we call them *Annotators* in our framework.

[1] via the JRuby implementation.

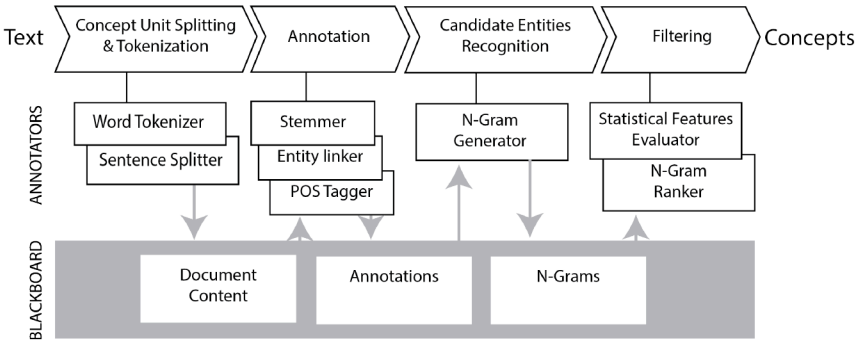

Figure 5.1: A Keyphrase Extraction pipeline built with the Distiller framework.

Implementing Knowledge Extraction tasks with Distiller ultimately is reduced to specifying a pipeline including the right annotators. Consider for instance the task of KP Extraction, usually such task is divided into the following steps: text pre-processing, candidate KP generation, and candidate KP selection and/or ranking. Distiller allows a quick deployment of such an application with the following annotators: a Sentence Splitter and a word Tokenizer to handle the pre-processing phase, a Stemmer, a POS Tagger and an optional Entity Linker to annotate the text, an N-Gram Generator to generate candidates, Scoring and Filtering modules to filter the most relevant candidates according to the annotations produced in the previous steps. The resulting pipeline is shown in Figure 5.1. Since each Annotator provides only a specific kind of knowledge, tailoring the pipeline to specific needs requires little effort. For instance, switching to another language requires replacing only the language dependant annotators, namely the POS Tagger, the Stemmer, and the Word Tokenizer. Other pipelines can be specified to implement different Knowledge Extraction and text mining tasks such as Sentiment Analysis, Summarization, or Authorship Identification.

5.1.2 Further Referential Space Model Applications

Aside from the related entities retrieval task described in Chapter 4, the Referential Space model presented in this work can be used in a number of other tasks.

Word Sense Disambiguation and Entity Linking, for instance, can be supported by a Referential Space model. Once the candidate anchors in a sentence or a paragraph are identified, all the pairwise distances between the possible interpretations of the candidate anchors are evaluated to find the interpretation that minimize the total distance between concepts. Finding the minimum distance could be seen as an application of the *Occam's razor* because, by doing so, we are selecting among competing hypotheses the one with the fewest assumptions. The main advantage of this solution with respect to other techniques such as the one used in [59] is that it is based more on semantics and less on syntactic considerations, allowing disambiguation and linking of entities that have little or no context. While most Entity

Linking techniques decide the most likely interpretation for an anchor by looking at the words that surround it, our approach looks at the concepts. By doing this some notoriously hard scenarios, like disambiguating entities mentioned in a list, rather than in a sentence, are easily solvable. Table 5.1 and 5.2 show some examples, provided by our working prototype, of Entity Linking in strings containing only a list of names, with no context other than the other items of the list. In the first query of the example presented in Table 5.1 the string "Panda" is correctly disambiguated, in an "animal" context, with the "Gian Panda" entity present in DBpedia. All the other strings are correctly identified as well. On the other hand, the same string "Panda", in the motor vehicle context of the second query, is matched with the DBpedia's entity "Fiat Panda". Even though the other strings of the second part of the first example, also exhibit a high polysemy, they are mutually disambiguated with plausible DBpedia entities. In the example shown in Table 5.2 a similar situation occurs. In the first query the string "Delphi" appears in a context related to programming languages. In that case, it is disambiguated with "Object Pascal", the entity of which it is the most famous incarnation. All the other strings are correctly associated with the programming language that they refer. In the second part of the example, where "Delphi" is enclosed in a Hellenic context, the string is rightly matched with a completely different entity: the archaeological site of Delphi in Greece. All the other strings are also disambiguated accordingly.

The most notable drawback of this technique is the large solution space it considers, implying a high number of comparisons that the system must perform to identify the minimal interpretation. As long as the number of candidate anchors is relatively small and the multiplicity of possible meanings for each candidate is low, computational times are acceptable, but for long texts with highly ambiguous terms, this might hinder the system's field usage. In our opinion this technique could be successfully employed paired with a more traditional one, based on word-level context: the more traditional techniques performs a first round of disambiguation, than anchors that cannot be resolved with a sufficiently high confidence, due to the absence of a proper context or to other factors, are handled with the Referential Space model. Experimentation is ongoing, however it appears evident that this hybrid solution could address several shortcomings of the current state of the art Named Entity Linking and Word Sense Disambiguation techniques.

Table 5.1: String disambiguation in two different queries.

	String	Entity
QUERY 1	Panda	http://dbpedia.org/resource/Giant_panda
	Bear	http://dbpedia.org/resource/Bear
	Weasel	http://dbpedia.org/resource/Weasel
QUERY 2	Panda	http://dbpedia.org/resource/Fiat_Panda
	Tesla	http://dbpedia.org/resource/Tesla_Roadster
	Leaf	http://dbpedia.org/resource/Nissan_Leaf

Table 5.2: Another example of string disambiguation in two different queries.

	String	Entity
QUERY 1	Delphi	http://dbpedia.org/resource/Object_Pascal
	C++	http://dbpedia.org/resource/C++
	Java	http://dbpedia.org/resource/Java_(programming_language)
	Python	http://dbpedia.org/resource/Python_(programming_language)
QUERY 2	Delphi	http://dbpedia.org/resource/Delphi
	Apollo	http://dbpedia.org/resource/Apollo
	Plutarch	http://dbpedia.org/resource/Plutarch

A

Complete Publications List

- Dario De Nart, Carlo Tasso, and Dante Degl'Innocenti. Users as crawlers: Exploiting metadata embedded in web pages for user profiling. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 22nd Conference on User Modeling, Adaptation, and Personalization co-located with the 22nd Conference on User Modeling, Adaptation, and Personalization (UMAP2014), Aalborg, Denmark, July 7-11, 2014.*, 2014

  In the last years we have witnessed the rapid growth of a broad range of Semantic Web technologies that have been successfully employed to enhance information retrieval, data mining and user experience in real-world applications. Several authors have proposed approaches towards ontological user modelling in order to address different issues of personalized systems, such as the cold start problem. In all of these works, non-structured data such as tags are matched, by means of various techniques, against an ontology in order to identify concepts and connections between them. However, due to recent popularity of semantic metadata formats such as microformats and RDFa, structured data are often embedded in many Web contents, with no need to "guess" them using a support ontology which may not be coherent with the actual content and the original goals of the author. In this paper we propose a novel approach towards ephemeral Web personalization based on extraction and enrichment of semantic metadata embedded in Web pages. The proposed system builds, at client-side, a rdf network that can be queried by a content provider in order to address personalized content.

  Main contribution: bibliographic research, system testing, dataset creation, paper review and corrections.

- Dario De Nart, Carlo Tasso, and Dante Degl'Innocenti. A thin-server approach to ephemeral web personalization exploiting RDF data embedded in web pages. In *Proceedings of the 2nd Workshop on Society, Privacy and the Semantic Web - Policy and Technology (PrivOn 2014) co-located with the 13th International Semantic Web Conference (ISWC 2014), Trento, Italy, October 20, 2014.*, 2014

Over the last years adaptive Web personalization has be-come a widespread service and all the major players of the WWW are providing it in various forms. Ephemeral personalization, in particular, deals with short time interests which are often tacitly entailed from user browsing behaviour or contextual information. Such personalization can be found almost anywhere in the Web in several forms, ranging from targeting advertising to automatic language localisation of content. In order to present personalized content a user model is typically built and maintained at server-side by collecting, explicitly or implicitly, user data. In the case of ephemeral personalization this means storing at server-side a huge amount of user behaviour data, which raises severe privacy concerns. The evolution of the semantic Web and the growing availability of semantic metadata embedded in Web pages allow a role reversal in the traditional personalization scenario. In this paper we present a novel approach towards ephemeral Web personalization consisting in a client-side semantic user model built by aggregating RDF data encountered by the user in his/her browsing activity and enriching them with triples extracted from DBpedia. Such user model is then queried by a server application via SPARQL to identify a user stereotype and finally address personalized content.

Main contribution: bibliographic research, system testing, dataset creation, paper review and corrections.

- Dario De Nart, Carlo Tasso, and Dante Degl'Innocenti. A semantic metadata generator for web pages based on keyphrase extraction. In *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014.*, pages 201–204, 2014

  The annotation of documents and web pages with semantic metatdata is an activity that can greatly increase the accuracy of Information Retrieval and Personalization systems, but the growing amount of text data available is too large for an extensive manual process. On the other hand, automatic keyphrase generation and wikification can significantly support this activity. In this demonstration we present a system that automatically extracts keyphrases, identifies candidate DBpedia entities, and returns as output a set of RDF triples compliant with the Opengraph and the Schema.org vocabularies.

  Main contribution: bibliographic research, system testing, dataset creation, paper review and corrections.

- Dante Degl'Innocenti, Dario De Nart, and Carlo Tasso. A new multi-lingual knowledge-base approach to keyphrase extraction for the italian language. In *KDIR 2014 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Rome, Italy, 21 - 24 October, 2014*, pages

78–85, 2014

Associating meaningful keyphrases to text documents and Web pages is an activity that can significantly increase the accuracy of Information Retrieval, Personalization and Recommender systems, but the growing amount of text data available is too large for an extensive manual annotation. On the other hand, automatic keyphrase generation can significantly support this activity. This task is already performed with satisfactory results by several systems proposed in the literature, however, most of them focuses solely on the English language which represents approximately more than 50% of Web contents. Only few other languages have been investigated and Italian, despite being the ninth most used language on the Web, is not among them. In order to overcome this shortage, we propose a novel multi-language, unsupervised, knowledge-based approach towards keyphrase generation. To support our claims, we developed DIKpE-G, a prototype system which integrates several kinds of knowledge for selecting and evaluating meaningful keyphrases, ranging from linguistic to statistical, meta/structural, social, and ontological knowledge. DIKpE-G performs well over English and Italian texts.

Main contribution: bibliographic research, system design, system development, system testing, dataset acquisition, experiment design, data analysis, paper writing.

- Dante Degl'Innocenti, Dario De Nart, and Carlo Tasso. A novel knowledge-based architecture for concept mining on italian and english texts. In Ana Fred, Jan L. G. Dietz, David Aveiro, Kecheng Liu, and Joaquim Filipe, editors, *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, volume 553 of *Communications in Computer and Information Science*, pages 132–142. Springer International Publishing, 2015

  In this paper we propose a novel knowledge-based, language independent, unsupervised approach towards keyphrase generation. We developed DIKpE-G, an experimental prototype system which integrates different kinds of knowledge, from linguistic to statistical, meta/structural, social, and ontological knowledge. DIKpE-G is capable to extract, evaluate, and infer meaningful concepts from a natural language text. The prototype performs well over both Italian and English texts.

  Main contribution: bibliographic research, system design, system development, system testing, dataset acquisition, experiment design, data analysis, paper writing.

- Dario De Nart, Dante Degl'Innocenti, Andrea Pavan, Marco Basaldella, and Carlo Tasso. Modelling the user modelling community (and other communities as well). In Francesco Ricci, Kalina Bontcheva, Owen Conlan, and Samus

Lawless, editors, *User Modeling, Adaptation and Personalization*, volume 9146 of *Lecture Notes in Computer Science*, pages 357–363. Springer International Publishing, 2015

Discovering and modelling research communities' activities is a task that can lead to a more effective scientific process and support the development of new technologies. Journals and conferences already offer an implicit clusterization of researchers and research topics, and social analysis techniques based on co-authorship relations can highlight hidden relationships among researchers, however, little work has been done on the actual content of publications. We claim that a content-based analysis on the full text of accepted papers may lead to a better modelling and understanding of communities' activities and their emerging trends. In this work we present an extensive case study of research community modelling based upon the analysis of over 450 events and 7000 papers.

Main contribution: bibliographic research, system testing, dataset creation, experiment execution, data analysis, paper review and corrections.

- Dario De Nart, Dante Degl'Innocenti, and Carlo Tasso. Introducing distiller: a lightweight framework for knowledge extraction and filtering. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modeling, Adaptation, and Personalization (UMAP 2015), Dublin, Ireland, June 29 - July 3, 2015.*, 2015

  Semantic content analysis is an activity that can greatly support a broad range of user modelling applications. Several automatic tools are available, however such systems usually provide little tuning possibilities and do not support integration with different systems. Personalization applications, on the other hand, are becoming increasingly multilingual and cross-domain. In this paper we present a novel framework for Knowledge Extraction, whose main goal is to support the development of new strategies and technologies and to ease the integration of the existing ones.
  Best Poster Paper Award

  Main contribution: bibliographic research, system testing, dataset creation, experiment execution, data analysis, paper review and corrections.

- Dario De Nart, Dante Degl'Innocenti, Marco Basaldella, and Carlo Tasso. A content-based approach to social network analysis: a case study on research communities. In *Proceedings of IRCDL 2015 - 11th Italian Research Conference on Digital Libraries, At Bozen-Bolzano, Italy.*, 2015

  Several works in literature investigated the activities of research communities using big data analysis, but the large majority of them focuses on papers and

co-authorship relations, ignoring that most of the scientific literature available is already clustered into journals and conferences with a well defined domain of interest. We are interested in bringing out underlying implicit relationships among such containers and in particular we are focusing on conferences and workshop proceedings available in open access and we exploit a semantic/conceptual analysis of the full free text content of each paper. We claim that such content-based analysis may lead us to a better understanding of the research communities' activities and their emerging trends. In this work we present a novel method for research communities activity analysis, based on the combination of the results of a Social Network Analysis phase and a Content-Based one. The major innovative contribution of this work is the usage of knowledge-based techniques to meaningfully extract from each of the considered papers the main topics discussed by its authors.

Main contribution: bibliographic research, system development, system testing, dataset creation, experiment execution, data analysis, paper review and corrections.

- Dario De Nart, Dante Degl'Innocenti, and Marco Peressotti. Well-stratified linked data for well-behaved data citation. *Bulletin of IEEE Technical Committee on Digital Libraries, Special Issue on Data Citation*, 2016
  In this paper we analyse the functional requirements of linked data citation and identify a minimal set of operations and primitives needed to realise such task. Citing linked data implies solving a series of data provenance issues and finding a way to identify data subsets. Those two tasks can be handled defining a simple type system inside data and verifying it with a type checker, which is significantly less complex than interpreting reified RDF statements and can be implemented in a non data invasive way. Finally we suggest that data citation should be handled outside of the data, possibly with an ad hoc language.

  Main contribution: bibliographic research, research idea development, paper review and corrections.

- Dante Degl'Innocenti, Dario De Nart, and Carlo Tasso. The importance of being referenced: Introducing referential semantic spaces. In *Proceedings of the Joint Second Workshop on Language and Ontologies (LangOnto2) & Terminology and Knowledge Structures (TermiKS), Workshop Abstracts, Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portoro, Slovenia, May 23, 2016.*, 2016

  The Web is constantly growing and to cope with its ever-increasing expansion semantic technologies are an extremely valuable ally. The major drawback of such technologies, however, is that providing a formal model of a domain is time consuming task that requires expert knowledge and, on the other hand,

extracting semantic data from text in an automatic way, although possible, is still extremely hard since it requires extensive human-annotated training corpora and non trivial document pre-processing. In this work we introduce a vector space representation of concept associations that can be built in an unsupervised way with minimal pre-processing effort and allows for associative reasoning supporting word sense disambiguation and related entity retrieval tasks.

Main contribution: bibliographic research, research idea, system design, system development, system testing, dataset acquisition, experiment design, experiment execution, data analysis, paper writing.

- Antonio D'Angelo and Dante Degl'Innocenti. Localization issues for an autonomous robot moving in a potentially adverse environment. In *Proceedings of the 14th International Conference on Intelligent Autonomous Systems (IAS 2016), Shanghai, China, July 3 - 7, 2016.*, 2016

  The aim of this paper is to face with the problem of localizing a robot during the navigation in a partially unknown environment. This feature becomes particularly noteworthy especially in the case of a colony of robots, possibly working with humans, inside a scenario where motion issues are crucial. Within this context the focus on self localization through GPS and INS/SINS integration overtakes merely questions about algorithm efficiency because self-localization is a relevant part of the task. Thus, unlike other approaches, we have focalised on this behavior as an attitude an autonomous system should enhance during the task execution. The tight coupling of GPS and INS sensors is understood as a mechanism which provides the autonomous robot with a refinement of INS use by comparing and/or adjusting the INS performance by exploiting the GPS-INS integration.

  Main contribution: bibliographic research, research idea development, system development, system testing, paper review and corrections.

- Eddy Maddalena, Marco Basaldella, Dario De Nart, Dante Degl'Innocenti, Stefano Mizzaro, and Gianluca Demartini. Crowdsourcing relevance assessments: The unexpected benefits of limiting the time to judge. In *Proceedings of the fourth Conference on Human Computation and Crowdsourcing (HCOMP 2016), Austin, Texas, 30 October 3 November, 2016.*, 2016

  Crowdsourcing has become an alternative approach to collect relevance judgments at scale thanks to the availability of crowdsourcing platforms and quality control techniques that allow to obtain reliable results. Previous work has used crowdsourcing to ask multiple crowd workers to judge the relevance of a document with respect to a query and studied how to best aggregate multiple judgments of the same topic-document pair. This paper addresses

an aspect that has been rather overlooked so far: we study how the time available to express a relevance judgment affects its quality. We also discuss the quality loss of making crowdsourced relevance judgments more efficient in terms of time taken to judge the relevance of a document. We use standard test collections to run a battery of experiments on the crowdsourcing platform CrowdFlower, studying how much time crowd workers need to judge the relevance of a document and at what is the effect of reducing the available time to judge on the overall quality of the judgments. Our extensive experiments compare judgments obtained under different types of time constraints with judgments obtained when no time constraints were put on the task. We measure judgment quality by different metrics of agreement with editorial judgments. Experimental results show that it is possible to reduce the cost of crowdsourced evaluation collection creation by reducing the time available to perform the judgments with no loss in quality. Most importantly, we observed that the introduction of limits on the time available to perform the judgments improves the overall judgment quality. Top judgment quality is obtained with 25-30 seconds to judge a topic-document pair.

Main contribution: research idea development, data cleaning, design of experiment 1, execution of experiment 1, data analysis of experiment 1, paper review and corrections.

- Dario De Nart, Dante Degl'Innocenti, Marco Peressotti, and Carlo Tasso. Stratifying semantic data for citation and trust: an introduction to rdfdf. In *Proceedings of IRCDL 2016 - 12th Italian Research Conference on Digital Libraries, At Florence-Firenze, Italy.*, 2016

  In this paper we analyse the functional requirements of linked data citation and identify a minimal set of operations and primitives needed to realise such task. Citing linked data implies solving a series of data provenance issues and finding a way to identify data subsets. Those two tasks can be handled defining a simple type system inside data and verifying it with a type checker, which is significantly less complex than interpreting reified RDF statements and can be implemented in a non data invasive way. Finally we suggest that data citation should be handled outside of the data, and propose a simple language to describe RDF documents where separation between data and meta information is explicitly specified.

  Main contribution: bibliographic research, research idea development, paper review and corrections.

- Muhammad Helmy, Dante Degl'Innocenti, Dario De Nart, and Carlo Tasso. Leveraging arabic morphology and syntax for better keyphrase extraction. In *Proceedings of the 20th International Conference on Asian Language Processing in Tainan, Taiwan, November 21-23, 2016.*, 2016

Arabic is one of the fastest growing languages on the Web, with an increasing amount of user generated content being published by both native and non-native speakers all over the world. Despite the great linguistic differences between Arabic and western languages such as English, most Arabic keyphrase extraction systems rely on approaches designed for western languages, thus ignoring its rich morphology and syntax. In this paper we present a new approach leveraging the Arabic morphology and syntax to generate a restricted set of meaningful candidates among which keyphrases are selected. Though employing a small set of well-known features to select the final keyphrases, our system consistently outperforms the well-known and established systems.

Main contribution: bibliographic research, research idea development, paper review and corrections.

- Dante Degl'Innocenti, Dario De Nart, Muhammad Helmy, and Carlo Tasso. Fast, accurate, multilingual semantic relatedness measurement using wikipedia links. In Khaled Shaalan, Aboul-Ella Hassanien, and M.F.Tolba, editors, *Intelligent Natural Language Processing: Trends and Applications*, Intelligent Systems Reference Library. Springer International Publishing, 2017

  In this chapter we present a fast, accurate, and elegant metric to assess semantic relatedness among entities included in an hypertextual corpus building an novel language independent Vector Space Model. Such a technique is based upon the Jaccard similarity coefficient, approximated with the MinHash technique to generate a constant-size vector fingerprint for each entity in the considered corpus. This strategy allows evaluation of pairwise semantic relatedness in constant time, no matter how many entities are included in the data and how dense the internal link structure is. Being semantic relatedness a subtle and somewhat subjective matter, we evaluated our approach by running user tests on a crowdsourcing platform. To achieve a better evaluation we considered two collaboratively built corpora: the English Wikipedia and the Italian Wikipedia, which differ significantly in size, topology, and user base. The evaluation suggests that the proposed technique is able to generate satisfactory results, outperforming commercial baseline systems regardless of the employed data and the cultural differences of the considered test users.

  Main contribution: bibliographic research, research idea development, paper review and corrections.

B

Keyphrase Extraction Quality Questionnaire

An example of the questionnaire provided to the users for the Italian KP extraction quality assessment task presented in Chapter 3 can be found down below. The questionnaire is structure as follows: on page 1 there is a detail description of the task, on page 2 the article's full text is presented, finally on page 3 the evaluation grid is shown.

Valuta la rilevanza di una lista di elementi rispetto a un testo

Panoramica

In questo esperimento vi verrà richiesto di leggere un testo e di valutare la qualità di una lista di elementi che cercano di catturarne l'argomento principale e che possono essere utilizzati per riassumerlo.

Passi

1. Leggi con attenzione il testo.
2. Leggi con attenzione gli elementi che compongono la lista.
3. Per ogni elemento della lista esprimi un solo giudizio, relativamente alla sua rilevanza rispetto al testo, barrando la relativa casella.

Valutazione

I giudizi che puoi esprimere per ogni elemento della lista sono i seguenti:

- Buono: l'elemento della lista è calzante rispetto al testo.
- Troppo Specifico: l'elemento è compreso nel testo ma non è utile agli scopi proposti. Ad esempio è un termine molto tecnico o una nozione accessoria del testo.
- Troppo Generico: L'elemento è troppo generale o comune rispetto all'argomento del testo.
- Incompleto: l'elemento è parziale, manca di una o più parole.
- Senza Senso: l'elemento non rappresenta un concetto.
- Non Rilevante: l'elemento non ricade nelle categorie precedenti ma comunque NON è utile ai fini proposti.

Testo: Terra, un pianeta abitabile grazie ai cicli geologici

L'emersione di roccia "fresca" con la formazione delle grandi catene montuose nel Cenozoico, 60 milioni di anni fa circa, ha contribuito in modo determinante all'equilibrio del ciclo del carbonio. La conclusione arriva da un nuovo studio che spiega almeno in parte perché il nostro è un pianeta goldilocks, dotato cioè di caratteristiche fisiche, chimiche e ambientali favorevoli allo sviluppo della vita.

La Terra è un pianeta abitabile perché gode di una miriade di condizioni fisiche chimiche e ambientali, dalla temperatura alla composizione dell'atmosfera, che si sono mantenute entro un intervallo di valori medi, favorevoli allo sviluppo e alla continuazione della vita. E questo si deve almeno in parte alle dinamiche geologiche che portano all'esposizione di nuovi strati di roccia durante la formazione delle grandi catene montuose e che hanno mantenuto in equilibrio il ciclo globale del carbonio e quindi un livello di anidride carbonica nell'atmosfera né troppo elevato né troppo basso.

E' quanto sostengono in un articolo sulla rivista "Nature" Mark Torres della University of Southern California a Los Angeles e colleghi dello stesso istituto e dell'Università di Nanjing, in Cina.

In termini teorici, quando in un sistema dinamico si determinano condizioni favorevoli solo se un certo numero di parametri si mantengono lontani dai valori estremi si parla di principio di Goldilocks, dal nome di una favola in cui la protagonista, una bambina dai riccioli biondi, si trova a suo agio nella casa degli orsi in cui è entrata di soppiatto solo quando trova un cibo "Né troppo freddo né troppo caldo", o un letto per dormire "né troppo grande né troppo piccolo".

Il principio è citato in diverse discipline, dall'ingegneria all'economia, dalla psicologia dello sviluppo alla biologia, ed è utilizzato in particolare dagli astrobiologi che cercano pianeti extrasolari che rientrino nelle cosiddette "zone abitabili", cioè né troppo vicini né troppo lontani dalla loro stella. In questo caso si parla, più sinteticamente e gergalmente, di pianeti goldilocks.

In quest'ultimo studio, Torres e colleghi hanno analizzato in particolare una delle fondamentali condizioni goldilocks della Terra, e cioè la concentrazione atmosferica di anidride carbonica, che si è sempre mantenuta entro valori compatibili con la vita.

La concentrazione di anidride carbonica è fortemente correlata ai processi geologici. La roccia "fresca", che emerge sulla superficie terrestre per esempio quando si formano le catene montuose, si comporta come una sorta di spugna, assorbendo grandi quantità di CO_2. Se questo processo non fosse stato controbilanciato, tutta l'anidride carbonica sarebbe stata assorbita entro pochi milioni di anni, facendo mancare l'effetto serra e rendendo la superficie del pianeta un luogo freddissimo e inadatto alla vita.

Una fonte di anidride carbonica atmosferica sono le eruzioni vulcaniche, ma recenti stime hanno evidenziato che il loro tasso di emissione di CO_2 non basta a bilanciare l'assorbimento da parte delle rocce.

Torres e colleghi hanno studiato alcuni campioni di roccia prelevati dalle Ande. Dalle analisi è emerso che il processo di meteorizzazione, cioè di disgregazione e alterazione chimica delle rocce affioranti in superficie dovuto al contatto con l'atmosfera, determina il rilascio di molta più anidride carbonica di quanto stimato in precedenza. In particolare, la rapida erosione nelle Ande porta alla luce molta pirite. La decomposizione chimica di questo minerale produce acidi che a loro volta determinano il rilascio di anidride carbonica da parte di altri minerali.

Questo importante riscontro ha portato a considerare le implicazioni globali del rilascio di CO_2 durante la formazione delle Ande, a partire da 60 milioni di anni fa, durante il periodo Cenozoico. Analizzando le registrazioni marine del ciclo del carbonio a lungo termine, gli studiosi hanno ricostruito l'equilibrio tra la produzione e l'assorbimento di anidride carbonica riconducibili al sollevamento di grandi catene montuose, concludendo che il rilascio di CO_2 per meteorizzazione potrebbe aver avuto un ruolo essenziale nella regolazione della concentrazione di anidride carbonica nell'atmosfera negli ultimi circa 60 milioni di anni, spiegando almeno in parte l'abitabilità del nostro pianeta.

Valutazione

Relativamente al testo letto, esprimi la qualità di ogni elemento della lista riportata qui in basso.
Gli elementi cercano di catturare l'argomento principale del testo e che possono essere utilizzati per riassumerlo.

	Buono	Troppo Specifico	Troppo Generico	Incompleto	Senza Senso	Non Rilevante
milioni						
catene						
anni						
carbonio						
catene montuose						
grandi catene						
anidride carbonica						
pianeta goldilocks						
milioni di anni						
ciclo del carbonio						
emersione di roccia						
equilibrio del ciclo						

3

Bibliography

[1] Lahsen Abouenour, Karim Bouzoubaa, and Paolo Rosso. On the evaluation and improvement of arabic wordnet coverage and usability. *Language resources and evaluation*, 47(3):891–917, 2013.

[2] Mohammed Attia, Antonio Toral, Lamia Tounsi, Pavel Pecina, and Josef van Genabith. Automatic extraction of arabic multiword expressions. In *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications (MWE 2010)*, pages 18–26, Beijing, China, August 2010. Association for Computational Linguistics.

[3] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853*, 2017.

[4] Arafat Awajan. Keyword extraction from arabic documents using term equivalence classes. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 14(2):7, 2015.

[5] Vít Baisa, Jan Michelfeit, Marek Medveď, and Milos Jakubicek. European union language resources in sketch engine. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA).

[6] Ken Barker and Nadia Cornacchia. Using noun phrase heads to extract document keyphrases. In *Advances in Artificial Intelligence*, pages 40–52. Springer, 2000.

[7] Marco Baroni, Silvia Bernardini, Federica Comastri, Lorenzo Piccioni, Alessandra Volpi, Guy Aston, and Marco Mazzoleni. Introducing the la repubblica corpus: A large, annotated, tei (xml)-compliant corpus of newspaper italian. *issues*, 2:5–163, 2004.

[8] Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226, 2009.

[9] Marco Basaldella, Dario De Nart, and Carlo Tasso. Introducing distiller: A unifying framework for knowledge extraction. In Stefano Ferilli and Nicola Ferro, editors, *IT@LIA@AI*IA*, volume 1509 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015.

[10] Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[11] Vladimír Benko. Aranea: Yet another family of (comparable) web corpora. In *International Conference on Text, Speech, and Dialogue*, pages 247–256. Springer, 2014.

[12] Vladimr Benko. Compatible sketch grammars for comparable corpora. In Andrea Abel, Chiara Vettori, and Natascia Ralli, editors, *Proceedings of the 16th EURALEX International Congress*, pages 417–430, Bolzano, Italy, jul 2014. EURAC research.

[13] Luisa Bentivogli and Emanuele Pianta. Exploiting parallel texts in the creation of multilingual semantically annotated resources: the multisemcor corpus. *Natural Language Engineering*, 11(03):247–261, 2005.

[14] Luisa Bentivogli, Emanuele Pianta, and Marcello Ranieri. MultiSemCor: an English Italian Aligned Corpus with a Shared Inventory of Senses. In *Proceedings of the Meaning Workshop*, 2005.

[15] Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.

[16] William Black, Sabri Elkateb, Horacio Rodriguez, Musa Alkhalifa, Piek Vossen, Adam Pease, and Christiane Fellbaum. Introducing the arabic wordnet project. In *Proceedings of the third international WordNet conference*, pages 295–300, 2006.

[17] Paolo Bouquet, Marc Ehrig, Jerome Euzenat, Enrico Franconi, Pascal Hitzler, Markus Krötzsch, Luciano Serafini, Giorgos Stamou, York Sure, and Sergio Tessaris. Specification of a common framework for characterizing alignment. Technical Report 2.2.1v2, University of Karlsruhe, December 2004.

[18] Eric Brill. A simple rule-based part of speech tagger. In *Proceedings of the workshop on Speech and Natural Language*, pages 112–116. Association for Computational Linguistics, 1992.

[19] A.Z. Broder. On the resemblance and containment of documents. In *Proceedings of Compression and Complexity of Sequences (SEQUENCES'97)*, pages 21–29. IEEE, June 1997.

[20] Steven Brown and Salvatore Attardo. *Understanding language structure, interaction, and variation: An introduction to applied linguistics and sociolinguistics for nonspecialists*. University of Michigan Press ELT, 2005.

[21] Alexander Budanitsky and Graeme Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 29–34, 2001.

[22] Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[23] Razvan C Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, volume 6, pages 9–16, 2006.

[24] Marine Carpuat, Grace Ngai, Pascale Fung, and Kenneth Church. Creating a bilingual ontology: A corpus-based approach for aligning wordnet and hownet. In *Proceedings of the 1st Global WordNet Conference*, 2002.

[25] Eugene Charniak. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 132–139. Association for Computational Linguistics, 2000.

[26] Hai Leong Chieu and Hwee Tou Ng. Named entity recognition: a maximum entropy approach using global information. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[27] Namyoun Choi, Il-Yeol Song, and Hyoil Han. A survey on ontology mapping. *ACM Sigmod Record*, 35(3):34–41, 2006.

[28] Rudi L Cilibrasi and Paul MB Vitanyi. The google similarity distance. *IEEE Transactions on knowledge and data engineering*, 19(3):370–383, 2007.

[29] Philipp Cimiano and Johanna Völker. text2onto. In *International Conference on Application of Natural Language to Information Systems*, pages 227–238. Springer, 2005.

[30] James Clarke, Vivek Srikumar, Mark Sammons, and Dan Roth. An nlp curator (or: How i learned to stop worrying and love nlp pipelines). In *LREC*, pages 3276–3283, 2012.

[31] Gianrenzo P Clivio and Marcel Danesi. *The sounds, forms, and uses of Italian: An introduction to Italian linguistics*. University of Toronto Press, 2000.

[32] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, volume 7, pages 708–716, 2007.

[33] Hamish Cunningham, Yorick Wilks, and Robert J Gaizauskas. Gate: a general architecture for text engineering. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 1057–1060. Association for Computational Linguistics, 1996.

[34] Antonio D'Angelo and Dante Degl'Innocenti. Localization issues for an autonomous robot moving in a potentially adverse environment. In *Proceedings of the 14th International Conference on Intelligent Autonomous Systems (IAS 2016), Shanghai, China, July 3 - 7, 2016.*, 2016.

[35] Marina Danilevsky, Chi Wang, Nihit Desai, Jingyi Guo, and Jiawei Han. KERT: automatic extraction and ranking of topical keyphrases from content-representative document titles. *CoRR*, abs/1306.0271, 2013.

[36] Ernesto D'Avanzo, Bernardo Magnini, and Alessandro Vallin. Keyphrase extraction for summarization purposes: The lake system at duc-2004. In *Proceedings of the 2004 document understanding conference*, 2004.

[37] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454. Genoa, 2006.

[38] Dario De Nart, Dante Degl'Innocenti, Marco Basaldella, and Carlo Tasso. A content-based approach to social network analysis: a case study on research communities. In *Proceedings of IRCDL 2015 - 11th Italian Research Conference on Digital Libraries, At Bozen-Bolzano, Italy.*, 2015.

[39] Dario De Nart, Dante Degl'Innocenti, Andrea Pavan, Marco Basaldella, and Carlo Tasso. Modelling the user modelling community (and other communities as well). In Francesco Ricci, Kalina Bontcheva, Owen Conlan, and Samus Lawless, editors, *User Modeling, Adaptation and Personalization*, volume 9146 of *Lecture Notes in Computer Science*, pages 357–363. Springer International Publishing, 2015.

[40] Dario De Nart, Dante Degl'Innocenti, and Marco Peressotti. Well-stratified linked data for well-behaved data citation. *Bulletin of IEEE Technical Committee on Digital Libraries, Special Issue on Data Citation*, 2016.

[41] Dario De Nart, Dante Degl'Innocenti, Marco Peressotti, and Carlo Tasso. Stratifying semantic data for citation and trust: an introduction to rdfdf. In *Proceedings of IRCDL 2016 - 12th Italian Research Conference on Digital Libraries, At Florence-Firenze, Italy.*, 2016.

[42] Dario De Nart, Dante Degl'Innocenti, and Carlo Tasso. Introducing distiller: a lightweight framework for knowledge extraction and filtering. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modeling, Adaptation, and Personalization (UMAP 2015), Dublin, Ireland, June 29 - July 3, 2015.*, 2015.

[43] Dario De Nart, Dante Degl'Innocenti, and Carlo Tasso. Introducing distiller: a lightweight framework for knowledge extraction and filtering. *UMAP 2015 Extended Proceedings*, 2015.

[44] Dario De Nart, Felice Ferrara, and Carlo Tasso. Personalized access to scientific publications: from recommendation to explanation. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 296–301. Springer, 2013.

[45] Dario De Nart and Carlo Tasso. A domain independent double layered approach to keyphrase generation. In *WEBIST 2014 - Proceedings of the 10th International Conference on Web Information Systems and Technologies*, pages 305–312. SCITEPRESS Science and Technology Publications, 2014.

[46] Dante Degl'Innocenti, Dario De Nart, Muhammad Helmy, and Carlo Tasso. Fast, accurate, multilingual semantic relatedness measurement using wikipedia links. In Khaled Shaalan, Aboul-Ella Hassanien, and M.F.Tolba, editors, *Intelligent Natural Language Processing: Trends and Applications*, Intelligent Systems Reference Library. Springer International Publishing, 2017.

[47] Dante Degl'Innocenti, Dario De Nart, and Carlo Tasso. A novel knowledge-based architecture for concept mining on italian and english texts. In Ana Fred, Jan L. G. Dietz, David Aveiro, Kecheng Liu, and Joaquim Filipe, editors, *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, volume 553 of *Communications in Computer and Information Science*, pages 132–142. Springer International Publishing, 2015.

[48] Dante Degl'Innocenti, Dario De Nart, and Carlo Tasso. The importance of being referenced: Introducing referential semantic spaces. In *Proceedings of the Joint Second Workshop on Language and Ontologies (LangOnto2) & Terminology and Knowledge Structures (TermiKS), Workshop Abstracts, Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portoro, Slovenia, May 23, 2016.*, 2016.

[49] Dante Degl'Innocenti, Dario De Nart, and Carlo Tasso. A new multi-lingual knowledge-base approach to keyphrase extraction for the italian language. In *KDIR 2014 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Rome, Italy, 21 - 24 October, 2014*, pages 78–85, 2014.

[50] Kristin Denham and Anne Lobeck. *Linguistics for everyone: An introduction*. Cengage Learning, 2012.

[51] Jörg Diederich and Wolf-Tilo Balke. The semantic growbag algorithm: Automatically deriving categorization systems. In *International Conference on Theory and Practice of Digital Libraries*, pages 1–13. Springer, 2007.

[52] Rehab Duwairi and Mona Hedaya. Automatic keyphrase extraction for arabic news documents based on kea system. *Journal of Intelligent and Fuzzy Systems*, 30(4):2101–2110, 2016.

[53] Samhaa R El-Beltagy and Ahmed Rafea. Kp-miner: A keyphrase extraction system for english and arabic documents. *Information Systems*, 34(1):132–144, 2009.

[54] Tarek El-Shishtawy and Abdulwahab Al-Sammak. Arabic keyphrase extraction using linguistic knowledge and machine learning techniques. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, 2009.

[55] Jérôme Euzenat. An api for ontology alignment. In *The Semantic Web–ISWC 2004*, pages 698–712. Springer, 2004.

[56] J. Fagan. Automatic phrase indexing for document retrieval. In *Proceedings of the 10th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '87, pages 91–101, New York, NY, USA, 1987. ACM.

[57] Ali Farghaly and Khaled Shaalan. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4):14, 2009.

[58] Rema Rossini Favretti, Fabio Tamburini, and Andrea Zaninello. Exploiting corpus evidence for automatic sense induction. In *Actas del III Congreso de la Asociación Española de Lingüística de Corpus*. Universitat Politècnica de València, 2011.

[59] Paolo Ferragina and Ugo Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1625–1628, New York, NY, USA, 2010. ACM.

[60] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and et al. Domain-specific keyphrase extraction. In *Proc. Sixteenth International Joint Conference on Artificial Intelligence*, pages 668–673. Morgan Kaufmann Publishers, 1999.

[61] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-manning. Domain specific keyphrase extraction. In *Proceedings of the 16th International Joint Conference on AI*, pages 668–673, 1999.

[62] V. Fromkin, R. Rodman, and N. Hyams. *An Introduction to Language*. Cengage Learning, 2013.

[63] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJcAI*, volume 7, pages 1606–1611, 2007.

[64] Risto Gligorov, Warner ten Kate, Zharko Aleksovski, and Frank Van Harmelen. Using google distance to weight approximate ontology matches. In *the 16th International Conference on World Wide Web*, pages 767–776. ACM, 2007.

[65] Nicola Grandi, Fabio Montermini, and Fabio Tamburini. Annotating large corpora for studying italian derivational morphology. *Lingue e linguaggio*, 10(2):227–244, 2011.

[66] Ilaria Grappasonno. Linguistica computazionale e applicazioni di interazione uomo-macchina. Master's thesis, Università degli Studi dell'Aquila, Dipartimento di Scienze Umane, A.A. 2013–14.

[67] B.F. Grimes, J.E. Grimes, and Summer Institute of Linguistics. *Ethnologue: Languages of the world*. Ethnologue. SIL International, 2000.

[68] Somya Gupta, Namita Mittal, and Alok Kumar. Rake-pmi automated keyphrase extraction: An unsupervised approach for automated extraction of keyphrases. In *Proceedings of the International Conference on Informatics and Analytics*, page 102. ACM, 2016.

[69] Nizar Y Habash. *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2010.

[70] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R Curran. Evaluating entity linking with wikipedia. *Artificial intelligence*, 194:130–150, 2013.

[71] Mounia Haddoud, Aïcha Mokhtari, Thierry Lecroq, and Saïd Abdeddaïm. Accurate keyphrase extraction from scientific papers by mining linguistic information. In *Proc. of the Workshop Mining Scientific Papers: Computational Linguistics and Bibliometrics, 15th International Society of Scientometrics and Informetrics Conference (ISSI), Istanbul, Turkey: http://ceur-ws. org*, 2015.

[72] Hannaneh Hajishirzi, Leila Zilles, Daniel S Weld, and Luke S Zettlemoyer. Joint coreference resolution and named-entity linking with multi-pass sieves. In *EMNLP*, pages 289–299, 2013.

[73] David Hall, Daniel Ramage, Jason Zaugg, Alexander Lehmann, Jonathan Merritt, Keith Stevens, Jason Baldridge, Timothy Hunter, Dave DeCaprio, Daniel Duckworth, et al. Scalanlp: Breeze, 2009.

[74] Sebastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, 8(1):1–254, 2015.

[75] Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

[76] Marti A Hearst. Automated discovery of wordnet relations. *WordNet: an electronic lexical database*, pages 131–153, 1998.

[77] Muhammad Helmy, Dante Degl'Innocenti, Dario De Nart, and Carlo Tasso. Leveraging arabic morphology and syntax for better keyphrase extraction. In *Proceedings of the 20th International Conference on Asian Language Processing in Tainan, Taiwan, November 21-23, 2016.*, 2016.

[78] Alfred Horn. On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic*, 16(1):1421, 1951.

[79] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[80] Tin Huynh, Kiem Hoang, Loc Do, Huong Tran, Hiep Luong, and Susan Gauch. Scientific publication recommendations based on collaborative citation networks. In *Collaboration Technologies and Systems (CTS), 2012 International Conference on*, pages 316–321. IEEE, 2012.

[81] Paul Jaccard. Lois de distribution florale. *Bulletin de la Socíeté Vaudoise des Sciences Naturelles*, 38:67–130, 1902.

[82] Matjaz Juršic, Igor Mozetic, Tomaz Erjavec, and Nada Lavrac. Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science*, 16(9):1190–1214, 2010.

[83] Su Nam Kim and Min-Yen Kan. Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proceedings of the workshop on multiword expressions: Identification, interpretation, disambiguation and applications*, pages 9–16. Association for Computational Linguistics, 2009.

[84] 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*, pages 21–26. Association for Computational Linguistics, 2010.

[85] J Peter Kincaid, Richard Braby, and John E Mears. Electronic authoring and delivery of technical information. *Journal of instructional development*, 11(2):8–13, 1988.

[86] Ekkehard Konig and Johan Van der Auwera. *The germanic languages*. Routledge, 2013.

[87] M. Krapivin, M. Marchese, A. Yadrantsau, and Yanchun Liang. Unsupervised key-phrases extraction from scientific papers using domain and linguistic knowledge. In *Digital Information Management, 2008. ICDIM 2008. Third International Conference on*, pages 105–112, Nov 2008.

[88] Vijay Krishnan and Christopher D Manning. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1121–1128. Association for Computational Linguistics, 2006.

[89] Tho Thi Ngoc Le, Minh Le Nguyen, and Akira Shimazu. Unsupervised keyword extraction for japanese legal documents. In *JURIX*, pages 97–106, 2013.

[90] Tho Thi Ngoc Le, Minh Le Nguyen, and Akira Shimazu. Unsupervised keyphrase extraction: Introducing new kinds of words to keyphrases. In *Australasian Joint Conference on Artificial Intelligence*, pages 665–671. Springer, 2016.

[91] Robert Leaman, Chih-Hsuan Wei, and Zhiyong Lu. tmchem: a high performance approach for chemical named entity recognition and normalization. *Journal of cheminformatics*, 7(1):1, 2015.

[92] Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics (ACL)*, 199.

[93] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Chris Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 2014.

[94] John Lehmann, Sean Monahan, Luke Nezda, Arnold Jung, and Ying Shi. Lcc approaches to knowledge base population at tac 2010. In *Proc. TAC 2010 Workshop*, 2010.

[95] J. Leskovec, A. Rajaraman, and J.D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2014.

[96] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

[97] Kenli Li, Wei Ai, Zhuo Tang, Fan Zhang, Lingang Jiang, Keqin Li, and Kai Hwang. Hadoop recognition of biomedical named entity using conditional random fields. *IEEE Transactions on Parallel and Distributed Systems*, 26(11):3040–3051, 2015.

[98] Marina Litvak, Mark Last, and Menahem Friedman. A new approach to improving multilingual summarization using a genetic algorithm. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 927–936. Association for Computational Linguistics, 2010.

[99] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 257–266, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[100] Patrice Lopez and Laurent Romary. Humb: Automatic key term extraction from scientific articles in grobid. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 248–251. Association for Computational Linguistics, 2010.

[101] Julie B Lovins. Development of a stemming algorithm. Technical report, DTIC Document, 1968.

[102] Verena Lyding, Egon Stemle, Claudia Borghetti, Marco Brunello, Sara Castagnoli, Felice Dell Orletta, Henrik Dittmann, Alessandro Lenci, and Vito Pirrelli. The PAISÀ Corpus of Italian Web Texts. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, pages 36–43, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

[103] Eddy Maddalena, Marco Basaldella, Dario De Nart, Dante Degl'Innocenti, Stefano Mizzaro, and Gianluca Demartini. Crowdsourcing relevance assessments: The unexpected benefits of limiting the time to judge. In *Proceedings of the fourth Conference on Human Computation and Crowdsourcing (HCOMP 2016), Austin, Texas, 30 October 3 November, 2016.*, 2016.

[104] Jayant Madhavan, Philip A Bernstein, AnHai Doan, and Alon Halevy. Corpus-based schema matching. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 57–68. IEEE, 2005.

[105] Alexander Maedche and Steffen Staab. Ontology learning. In *Handbook on ontologies*, pages 173–190. Springer, 2004.

[106] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[107] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[108] Luis Marujo, Anatole Gershman, Jaime Carbonell, Robert Frederking, and Jo ao P. Neto. Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization. In *Proceedings of LREC 2012*. ELRA, 2012.

[109] Luis Marujo, Márcio Viveiros, and Jo ao P. Neto. Keyphrase Cloud Generation of Broadcast News. In *Proceedings of Interspeech 2011*. ISCA, September 2011.

[110] Yutaka Matsuo and Mitsuru Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01):157–169, 2004.

[111] Paul McNamee, Hoa Trang Dang, Heather Simpson, Patrick Schone, and Stephanie Strassel. An evaluation of technologies for knowledge base population. In *LREC*, 2010.

[112] Olena Medelyan, Eibe Frank, and Ian H. Witten. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1318–1327, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[113] Olena Medelyan and Ian H Witten. Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 296–297. ACM, 2006.

[114] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.

[115] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.

[116] M. Minsky. A framework for representing knowledge. In P. Winston, editor, *The Psychology of Computer Vision*, chapter 6, pages 211–277. McGraw-Hill, New York, 1975.

[117] Cristina Monti, Claudio Bendazzoli, Annalisa Sandrelli, and Mariachiara Russo. Studying directionality in simultaneous interpreting through an electronic corpus: Epic (european parliament interpreting corpus). *Meta*, 50(4), 2005.

[118] Andrea Moro, Francesco Cecconi, and Roberto Navigli. Multilingual word sense disambiguation and entity linking for everybody. In *Proceedings of the 2014 International Conference on Posters & Demonstrations Track-Volume 1272*, pages 25–28. CEUR-WS. org, 2014.

[119] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: A unified approach. *Transactions of the Association for Computational Linguistics*, 2, 2014.

[120] Adrian Muller, Jochen Dorre, Peter Gerstl, and Roland Seiffert. The taxgen framework: Automating the generation of a taxonomy for a large document collection. In *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, pages 9–pp. IEEE, 1999.

[121] Cataldo Musto, Pasquale Lops, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro. Semantics-aware graph-based recommender systems exploiting linked open data. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, pages 229–237. ACM, 2016.

[122] Mahmoud Nabil, AF Atiya, and M Aly. New approaches for extracting arabic keyphrases. In *Proceedings of First International Conference on Arabic Computational Linguistics (ACLing)*, pages 133–137. IEEE, 2015.

[123] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[124] Dario De Nart, Carlo Tasso, and Dante Degl'Innocenti. A semantic metadata generator for web pages based on keyphrase extraction. In *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014.*, pages 201–204, 2014.

[125] Dario De Nart, Carlo Tasso, and Dante Degl'Innocenti. A thin-server approach to ephemeral web personalization exploiting RDF data embedded in web pages. In *Proceedings of the 2nd Workshop on Society, Privacy and the Semantic Web - Policy and Technology (PrivOn 2014) co-located with the 13th*

International Semantic Web Conference (ISWC 2014), Trento, Italy, October 20, 2014., 2014.

[126] Dario De Nart, Carlo Tasso, and Dante Degl'Innocenti. Users as crawlers: Exploiting metadata embedded in web pages for user profiling. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 22nd Conference on User Modeling, Adaptation, and Personalization co-located with the 22nd Conference on User Modeling, Adaptation, and Personalization (UMAP2014), Aalborg, Denmark, July 7-11, 2014.*, 2014.

[127] Grace Ngai, Marine Carpuat, and Pascale Fung. Identifying concepts across languages: A first step towards a corpus-based approach to automatic ontology alignment. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[128] Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. J-nerd: Joint named entity recognition and disambiguation with rich linguistic features. *Transactions of the Association for Computational Linguistics*, 4:215–229, 2016.

[129] Thuy Dung Nguyen and Min-Yen Kan. Keyphrase extraction in scientific publications. In *International Conference on Asian Digital Libraries*, pages 317–326. Springer, 2007.

[130] Joseph D. Novak. *Learning, creating, and using knowledge: Concept maps as facilitative tools in schools and corporations.* Taylor & Francis, London, United Kingdom, 2010.

[131] Francesco Osborne and Enrico Motta. Inferring semantic relations by user feedback. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 339–355. Springer, 2014.

[132] Francesco Osborne and Enrico Motta. Klink-2: integrating multiple web sources to generate semantic topic networks. In *International Semantic Web Conference*, pages 408–424. Springer, 2015.

[133] Chris D. Paice. Another stemmer. *SIGIR Forum*, 24(3):56–61, November 1990.

[134] F. Parker, K. Riley, and K.L. Riley. *Linguistics for Non-linguists.* Allyn & Bacon, 2010.

[135] Arfath Pasha, Mohamed Al-Badrashiny, Mona T Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *LREC*, volume 14, pages 1094–1101, 2014.

[136] Mari-Sanna Paukkeri, Ilari T Nieminen, Matti Pöllä, and Timo Honkela. A language-independent approach to keyphrase extraction and evaluation. In *COLING (Posters)*, pages 83–86. Citeseer, 2008.

[137] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440. Association for Computational Linguistics, 2006.

[138] Mohammad Taher Pilehvar and Roberto Navigli. From senses to texts: An all-in-one graph-based approach for measuring semantic similarity. *Artificial Intelligence*, 228:95–128, 2015.

[139] Martin F Porter. Snowball: A language for stemming algorithms, 2001.

[140] Nirmala Pudota, Antonina Dattolo, Andrea Baruzzo, Felice Ferrara, and Carlo Tasso. Automatic keyphrase extraction and ontology mining for content-based tag recommendation. *Int. J. Intell. Syst.*, 25(12):1158–1186, December 2010.

[141] Nirmala Pudota, Antonina Dattolo, Andrea Baruzzo, and Carlo Tasso. A new domain independent keyphrase extraction system. In *Italian Research Conference on Digital Libraries*, pages 67–78. Springer, 2010.

[142] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics, 2009.

[143] Giuseppe Rizzo and Raphaël Troncy. Nerd: a framework for unifying named entity recognition and disambiguation extraction tools. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 73–76. Association for Computational Linguistics, 2012.

[144] M Andrea Rodríguez and Max J. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *IEEE transactions on knowledge and data engineering*, 15(2):442–456, 2003.

[145] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. *Text Mining*, pages 1–20, 2010.

[146] Adriana Roventini, Antonietta Alonge, Nicoletta Calzolari, Bernardo Magnini, and Francesca Bertagna. Italwordnet: a large semantic database for italian. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC 2000)*, pages 783–790, 2000.

[147] G. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[148] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[149] Mark Sanderson and Bruce Croft. Deriving concept hierarchies from text. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–213. ACM, 1999.

[150] Helmut Schmid. Improvements in part-of-speech tagging with an application to german. In *In proceedings of the acl sigdat-workshop*. Citeseer, 1995.

[151] Helmut Schmid. Treetagger— a language independent part-of-speech tagger. *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*, 43:28, 1995.

[152] Alexander Thorsten Schutz. *Keyphrase Extraction from Single Documents in the Open Domain Exploiting Linguistic and Statistical Methods.* National University of Ireland, Galway, 2008.

[153] John Scott and Gordon Marshall. *A Dictionary of Sociology.* Oxford University Press, 2009.

[154] Pavel Shvaiko and Jérôme Euzenat. Ontology matching: state of the art and future challenges. *Knowledge and Data Engineering, IEEE Transactions on*, 25(1):158–176, 2013.

[155] J. F. Sowa. Conceptual graphs for a data base interface. *IBM Journal of Research and Development*, 20(4):336–357, July 1976.

[156] John F Sowa. Semantic networks. *Encyclopedia of Cognitive Science*, 2006.

[157] Sebastian Spiegler. Statistcs of the common crawl corpus 2012. Technical report, Technical report, SwiftKey, 2013.

[158] Ralf Steinberger, Mohamed Ebrahim, Alexandros Poulis, Manuel Carrasco-Benitez, Patrick Schlüter, Marek Przybyszewski, and Signe Gilbro. An overview of the european unions highly multilingual parallel corpora. *Language resources and evaluation*, 48(4):679–707, 2014.

[159] Ralf Steinberger, Andreas Eisele, Szymon Klocek, Spyridon Pilos, and Patrick Schlüter. Dgt-tm: A freely available translation memory in 22 languages. In *Proceedings of the 8th international conference on Language Resources and Evaluation (LREC 2012)*, Istanbul, may 2012.

[160] Xu Sun. Structure regularization for structured prediction. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2402–2410. Curran Associates, Inc., 2014.

[161] Maksim Tkachenko and Andrey Simanovsky. Named entity recognition: Exploring features. In *KONVENS*, pages 118–127, 2012.

[162] Kristina Toutanova and Christopher D Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics, 2000.

[163] Peter D Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, 2000.

[164] Peter D. Turney. Coherent keyphrase extraction via web mining. In *In Proceedings of IJCAI*, pages 434–439, 2003.

[165] Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January 2010.

[166] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Daniel Gerber, Sandro Athaide Coelho, Sören Auer, and Andreas Both. Agdistis-graph-based disambiguation of named entities using linked data. In *The Semantic Web–ISWC 2014*, pages 457–471. Springer, 2014.

[167] Cornelis J Van Rijsbergen, Stephen Edward Robertson, and Martin F Porter. *New models in probabilistic information retrieval*. British Library Research and Development Department, 1980.

[168] L. Vannini, H.L. Crosnier, and MAAYA. *Net.lang: Towards the Multilingual Cyberspace*. C & F, 2012.

[169] Vasudeva Varma, Praveen Bysani, Kranthi Reddy, Vijay Bharath Reddy, Sudheer Kovelamudi, Srikanth Reddy Vaddepally, Radheshyam Nanduri, N Kiran Kumar, Santhosh Gsk, and Prasad Pingali. Iiit hyderabad in guided summarization and knowledge base population. *International Institute of Information Technology*, 2010.

[170] Nina Wacholder, Yael Ravin, and Misook Choi. Disambiguation of proper names in text. In *Proceedings of the fifth conference on Applied natural language processing*, pages 202–208. Association for Computational Linguistics, 1997.

[171] Xiaojun Wan and Jianguo Xiao. Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of 22nd International Conference on Computational Linguistics*, pages 969–976, Manchester, UK, 2008.

[172] Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. *arXiv preprint arXiv:1408.2927*, 2014.

[173] Julie Weeds and David Weir. Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics*, 31(4):439–475, 2005.

[174] Ian Witten and David Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, pages 25–30, 2008.

[175] 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*, pages 254–255. ACM, 1999.

[176] Gerhard Wohlgenannt, Albert Weichselbraun, Arno Scharl, and Marta Sabou. Dynamic integration of multiple evidence sources for ontology learning. *Journal of Information and Data Management*, 3(3):243, 2012.

[177] Chengzhi Zhang. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*, 4(3):1169–1180, 2008.