TM-Gen: A Topic Map Generator from Text Documents

Angel L. Garrido, María G. Buey, Sandra Escudero, Sergio Ilarri, Eduardo Mena
IIS Department
University of Zaragoza
Zaragoza, Spain
Email: {***garrido, mgbuey, sandra.escudero, silarri, emena***}***@unizar.es***

Sara B. Silveira
Computer Department
University of Lisbon
Lisbon, Portugal
Email: ***sara.silveira@di.fc.ul.pt***

***Abstract*—The vast amount of text documents stored in digital format is growing at a frantic rhythm each day. Therefore, tools able to find accurate information by searching in natural language information repositories are gaining great interest in recent years. In this context, there are especially interesting tools capable of dealing with large amounts of text information and deriving human-readable summaries. However, one step further is to be able not only to summarize, but to extract the knowledge stored in those texts, and even represent it graphically.**

In this paper we present an architecture to generate automatically a conceptual representation of knowledge stored in a set of text-based documents. For this purpose we have used the topic maps standard and we have developed a method that combines text mining, statistics, linguistic tools, and semantics to obtain a graphical representation of the information contained therein, which can be coded using a knowledge representation language such as RDF or OWL. The procedure is language-independent, fully automatic, self-adjusting, and it does not need manual configuration by the user. Although the validation of a graphic knowledge representation system is very subjective, we have been able to take advantage of an intermediate product of the process to make an experimental validation of our proposal.

***Keywords*-Knowledge acquisition; text mining; ontologies; topic maps; linguistics.**

I. Introduction

Nowadays, any organization owns text document collections in digital format that are constantly growing. This, coupled with the explosion of written information generated through the Internet leads us to have a huge amount of text information, so vast that it often takes a great effort to find what is really needed since most of it is hidden, messy and unsorted. So, proposals related to improving information and knowledge retrieval tools are receiving a great interest.

In the context of information systems, the usual way of representing knowledge is by using ontologies. An ontology is defined as a formal and explicit specification of a shared conceptualization [1] and it can be stored in various digital formats suitable for machines, so it is often desirable to have a tool that shows knowledge through a schema which is easier to interpret by humans.

There are different types of schemas for knowledge representation, but we have focused our attention on topic maps [2], since their simplicity makes them understandable by anyone, and besides they have a large semantic expressiveness, which allows us to store and transmit knowledge [3].

Our proposal is an automatic system called TM-GEN ("Topic Map GENerator"), capable of generating a topic map from scratch, relying on a set of input texts, and using tools without any information related to the context of the documents, in order to make the system as automatic as possible. The experimental results obtained after working with Spanish texts belonging to a news corpus from the newspaper *Heraldo de Aragón*[1] are very promising and show the interest of the proposal.

Therefore, this paper provides two main contributions:

- Firstly, we present a new architecture valid for any knowledge extraction task from unstructured text-based information, without using any context information.
- Secondly, we have faced the problem of the evaluation of topic maps by proposing a new objective method.

This paper is structured as follows. Section II describes the state of the art. Section III explains the general architecture of our solution and presents the proposed algorithm. Section IV discusses the results of our experiments with real data. Finally, Section V provides our conclusions and some lines of future work.

II. Related Work

In this section we describe the state of the art by reviewing the general aspects of topic maps, listing some existing tools, and introducing the issue of automatic summaries.

A. Context

A concept map is a diagram that shows relationships between concepts within a context. The concepts are represented in a hierarchical graph with the most inclusive and most relevant concepts at the top of the map, and the more specific and less relevant concepts below. Concept maps facilitate sense-making and learning by the individuals who make them, and by those who use them, because they are constructed to reflect the organization of the memory

[1] http://www.heraldo.es/

system. Concept maps are used in various fields, such as Biology [4], or Engineering [5]. Novak and Musonda claimed that concept maps improve learning and teaching, as they facilitate extracting knowledge and representing it graphically [6].

There is a set of standard rules to manage the representation and exchange of knowledge. The standard ISO is known as ISO/IEC 13250:2003, or more commonly as XTM (XML Topic Maps) [2]. A topic map in XML is composed of topics, associations, and occurrences. Topics are the elements of the topic map, associations are the relations among the different topics, and occurrences are the appearances of each topic in the texts. Topic maps are similar to concept maps in several aspects, although only topic maps are standardized.

B. Topic map tools

There are many tools for building automatically topic maps, for example [7], which perform an automatic extraction of topic map from web pages by crawling, through the mining of information about the topics and the relationships present in the web pages. These approaches apply heuristics for extracting this information from web sites, such as statistical and linguistic analysis, and they use annotation for the automatic extraction of linguistic entities. Other studies, like [8], propose a semi-automatic generation incorporating machine learning techniques in the process.

Although many methods for concept extraction from texts and many tools to generate topic maps have been proposed, most of them do not take so much into account the semantic relations between the topics and the associations. They usually apply semi-automatic processes that need machine learning techniques and preprocessing, but we miss in these works the fact of taking into account the semantic aspects and working them in more detail. This point is one of the distinguishing features of our work.

C. Automatic summaries

In some ways, the creation of a topic map from a document also relates to the preparation of a summary in text format. After all, in both cases, the goal is to collect the key ideas of a text and rewrite it synthetically with a new format. We have exploited this similarity for the construction and evaluation of the topic maps.

Regarding the creation of automatic summaries, it is important to mention classical works that use statistical criteria to detect sentences that contain high-frequency terms [9]. Other recent works use positional criteria [10]. Due to our interest in working with multi-documents, we have analyzed other similar works, for example [11], that use domain-independent techniques based mainly on fast statistical processing, a metric for reducing redundancy and maximizing diversity in the selected passages or that use a cluster centroid with techniques such as graph matching, maximal marginal relevance, and language generation. Also we find very interesting the recent contributions of SIMBA [12], which has a smart procedure to simplify sentences to ensure the compression of the summary. To carry out this task, SIMBA applies a two-stage process of clusterization: clustering sentences by similarity and clustering sentences by keyword.

III. Methodology

Our purpose is to create a fully automated system able to extract information and knowledge from a set of texts and collect that information in a topic map. To do its work, the first task is preprocessing the texts to find relevant information that the process needs later. Then, each text is processed separately to obtain its own topic map. This is done by dividing the text into sentences with the purpose of analyzing them separately and assigning them a relevance score, in order to find those that are most important in the text (i.e., they will give us more information and knowledge). Afterwards, TM-Gen analyzes syntactically the sentences to find the best candidates to be a topic, and then the system establishes associations between them. Next, TM-Gen carries out a semantic simplification in case there exist redundancies, incompatible associations, or ambiguities. Finally, when all the texts have been analyzed and their corresponding topic map is generated, the process merges them all into a single topic map.

A. Process pipeline

In this subsection we describe the pipeline of the process in detail:

1) *Preprocessing*: TM-Gen performs an analysis of the texts to obtain information related to their words, that is required in later tasks. To do this, we use an NLP tool to extract and lemmatize the words that constitute the texts, and to identify named entities present in them. The extracted information is stored in two catalogs:
   - A list of named entities [13] identified by the natural language processing tool, that correspond to names of people, places, organizations, etc. Before storing them, during the preprocessing task these named entities are assigned with a weight because they are relevant information present in a text. In this phase is where TM-Gen uses a gazetteer [14] to identify which named entities are locations.
   - A list of frequencies of the words present in the texts. The process uses a method that calculates statistics from words lemmatized by the NLP tool and included in the texts and calculates their frequencies.
2) *Extraction of keywords*: It extracts a list of keywords from each text using the well-known TF-IDF (Term

Frequency - Inverse Document Frequency) [15]. Later, these words allow us to identify which parts of the text provide more knowledge. In this phase we use the frequencies of the words extracted in the previous preprocessing task.

3) *Extraction of Named Entities*: TM-Gen extracts a list of named entities from each text by using an NLP tool. These named entities are important candidates to be concepts in the topic map, as they provide relevant information and each of them constitutes a concept itself. It also takes into account the frequencies of these named entities extracted in the preprocessing phase.
4) *Text split in sentences*: It divides each text into sentences to identify later which of those sentences have more relevant information.
5) *Sentences scoring*: TM-Gen uses a method to score the sentences in order to identify which ones are the most relevant. For this task, it takes into account the keywords and named entities that appear in a sentence and it scores each one using the frequencies of the words extracted in the preprocessing step. This method is described in more detail in Section III-C.
6) *Sort sentences*: Once the sentences have been scored, they are ordered from the highest relevance to the lowest. At this point, the set of the sentences that are placed higher on the list (i.e., with higher scores) constitutes already a summary of the text.
7) *Sentence analysis*: The process performs a syntactic and grammatical analysis of each sentence in order to identify the function of each word in the sentence and its type. For this purpose we use a parser tool [16]. It is important to extract this information from the sentences because the best candidates to be concepts in the topic map are the words whose function is to be the subject of the sentence; a subject is typically a noun, noun phrase, or pronoun. It is also important to identify verbs in the sentence, as they establish associations between topics.
8) *Addition of topics*: TM-Gen adds to the topic map the concept candidates found in the previous step. These candidates are the subjects identified. For each candidate, if it has been previously included in the topic map, then the topic is not added but it will be assigned with a higher weight. The intuition is that the topics that are repeated more than once in several sentences tend to have more relevance in the text and they usually provide more information.
9) *Addition of associations*: TM-Gen adds to the topic map the associations between topics found in each sentence. These associations are given by the verbs present in the sentence. TM-Gen performs this task by searching the subject included as topic, and then it adds the verb as its association. Finally, it links its verb complement with the topic and with the association as a new topic.
10) *Semantic simplification*: Once topics and associations are added into the topic map, the process performs a semantic analysis of them and it makes a simplification of the topic map in case it finds redundancies, incompatible associations, or ambiguities. This task is described in more detail in Section III-D.
11) *Evaluation*: Each time the system adds elements corresponding to a sentence into a topic map, an evaluation process is performed. In this process it checks if the number of topics added is enough, and if so it stops processing sentences and it continues with the next task. This number can be adjusted according to the desired depth level of the topic map.
12) *Validation*: The process checks if the topic map has been built correctly according to the standards for topic maps [2], and it also searches for possible mistakes generated in previous phases of the process and fixes them.
13) *Merge*: Once the topic map for each text in the input set has been generated, TM-Gen performs a merging of all of them into a single topic map. To do this, we use a method previously developed in [17]. The method for merging is called SIM (Subject Identity Measure) and it is responsible for describing the relation among two subjects or topics.
14) *Creation of concept maps and ontologies*: The process of design and creation of ontologies can be a pretty complex process. For this reason, we suggest the utilization of a representation that can be integrated with ontologies, so these can be obtained automatically. For the conversion from topic maps to ontologies we need a concept map in the XTM language to create a corresponding file in the OWL language [18]. Furthermore, to perform the graphic representation of topic maps we use ONTOPIA [19], as it automatically draws topic maps from an XTM file.

B. Algorithm

The next algorithm that is detailed below in pseudo-code explains the process detailed in the previous sections:

```
PROGRAM TMGen (
        INPUT: g as Gazetteer,
               LDB as LexicalDB,
               txt_list as ListOf(string);
        OUTPUT: tm_ok as topic_map);
BEGIN
   tm_list := empty_list;
   Preprocessing(txt_list);
   FOR EACH text IN txt_list DO
       tm := new topic_map;
       kw_list := GetKeywords(text);
       ne_list := GetNamedEntities(text,g);
       sentence_list := SplitText(text);
       FOR EACH s IN sentence_list DO
```

```
            ScoreSentence(s,kw_list,ne_list);
        END FOR
        SortSentences(sentence_list);
        WHILE NOT fit(topic_map) DO
            s := sentence_list.next;
            c := TopicAnalysis(s);
            assoc := AssocAnalysis(s);
            tm := AddTopic(c, tm);
            tm := AddAssoc(assoc, tm);
            SemanticSimplif(tm,LDB);
        END WHILE
        Validate(tm);
        tm_list.add(tm);
    END FOR
    tm_ok := Merge(tm_list);
    Output := tm_ok;
END.
```

C. Determining the relevant sentences

The proposed method searches in a sentence the words that match with keywords and named entities that appear in the entire text. We use an approximation of the method proposed in [12].

The sentence scoring procedure defines the sentence relevance in the overall collection of sentences obtained from each input text, and it is the sum of the `tf-idf` scores ($tfidf_w$) [15] (computed considering the word lemmas obtained previously) of each word of the sentence s, smoothed by the number of words in the sentence (*totW*). This metric states that the relevance of a sentence not only depends on the frequency of the words present in it, but also on the number of texts in which the words appear. Equation 1 describes the computation of this score:

$$score_s = \frac{\sum_w (tfidf_w)}{totW} \quad (1)$$

The next stages of processing aim at identifying relevant information in the collection of sentences, in two steps:

1) *Similarity clustering*: In order to identify sentences conveying the same information, the process clusters them considering their degree of similarity. The similarity between two sentences comprises two dimensions, computed considering the word lemmas: the sentences subsequences and the word overlap. These two dimensions are combined in a similarity value. If the similarity value is higher than the similarity threshold, the sentences are grouped in the same cluster. The sentence with the highest score from each cluster is selected to be used in the next step.
2) *Keyword clustering*: The algorithm that clusters sentences by keywords is an adapted version of the K-means algorithm [20]. Keyword clustering groups sentences based on the occurrence of the keywords. The keywords extracted previously represent the clusters. A sentence is added to the cluster represented by the keyword that occurs more often in it. The sentences that do not contain any keywords are ignored.

The next task of the process orders the sentences based on their score after both clustering phases have been executed, defining the order of the sentences to be processed in the next tasks. These sentences are considered more significant than the others, since they address the main topics conveyed by the collection of sentences.

D. Semantic simplification

Most methods and techniques developed to extract concept maps from texts usually take into account syntactic and grammatical information of the words that compose a sentence (or a text), and statistical analysis of the most relevant words, but there are very few proposals that exploit semantic aspects. Thus, it is possible that the topic map generated includes topic redundancies. This means that there could exist several topics that relate to the same concept, i.e., they are synonyms. At this task, TM-Gen conducts an analysis to search this type of redundancies and, in case of finding them, removes and reduce them into a single concept, using for this purpose a lexical database that contains semantic information of the words of a language. For example, if the process has generated a topic map where the topics "objective" and "purpose" appear, this method will search and select their best meaning and reduce them into only one topic, gathering and connecting their associations with it.

At the same time, the process can identify ambiguities in the topics. Thus, two or more topics can have the same meaning in a certain context but not in others. In this case, we use a disambiguation engine [21] taking into account associations and topics that are close to those concepts in the topic map hierarchy. If the process finds that they are synonyms, then it reduces them in the way described above.

Moreover, the process is also responsible for identifying the most general concepts, in order to extend and generalize the knowledge content extracted. For example, if it finds "kids" in the topic map it substitutes the word with "children", which is more general and formal. At this phase, TM-Gen also recognizes the possible incompatible associations between topics. This occurs when two completely opposite associations leave or arrive at a topic, or when an association does not share a relation with a topic. In this case, it tries to solve it by using the disambiguation engine or remove them if it does not find a solution. This method helps to correct errors derived from the previously mentioned simplification or errors in the text itself.

IV. Experimental evaluation

This section discusses the results of experimental tests carried out to check the performance of the proposed algorithm. For testing in a real environment we have used a corpus of 4,433 news of *Heraldo de Aragón*, a major Spanish media, containing sets of texts in Spanish and its

corresponding summaries made by the professional documentation department of the newspaper. The purpose of our system is to approximate the automatic generation of knowledge to those professional summaries. To evaluate this, we used an intermediate product of our process: the natural language summary used as a basis to construct the topic map of each text. So, we will compare our automatic summary to the abstract elaborated by expert humans.

We compared our summaries with the summaries elaborated by an external tool: SweSum[2] [22]. In our system, we have used Freeling [23] as Spanish NLP tool, both for morphological analysis as for syntactic analysis of the texts. As a lexical database we have used EuroWordNet [24] concerning the evaluation itself. After both summaries (the one generated by SweSum and the one obtained by our system) had been built for each input, they were compared with the corpus of ideal summaries using ROUGE [25], which is a package for automatic evaluation of summaries.

The results of the experiments are shown in Fig. 1.

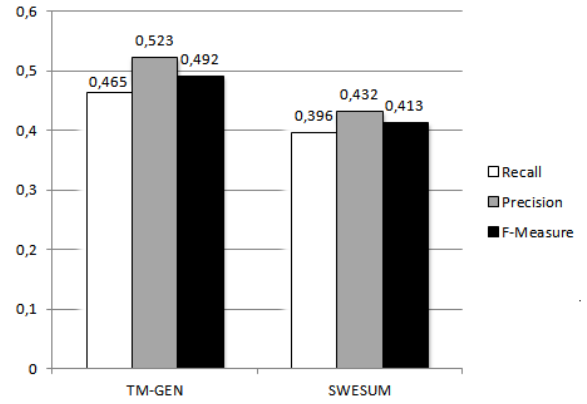

Figure 1. ROUGE-L metrics for TM-Gen and Swesum summaries

We analyze the following measures, commonly used in the Information Retrieval context [26]: the precision, the recall, and the F-Measure. By observing Fig. 1, we can see that TM-Gen has a better performance than SweSum. The F-Measure value for TM-Gen overcomes the one of SweSum in eight percentage points, meaning that TM-Gen summaries have more significant information than the ones of SweSum. The precision value obtained by TM-Gen is very interesting. A high precision value means that, considering all the information in the input texts, the retrieved information is relevant. Thus, obtaining the most relevant information in the sentences by discarding their less relevant data, which is ignored during the construction of the topic map, ensures that the summary contains indeed the most important information conveyed by each of its sentences. The recall values of the two systems are closer than the ones concerning precision. Thus, considering the evaluation results, we can conclude that this approach produces better summaries when compared to the one used by SweSum.

Regarding the graphical representation, the impression is that the topic maps generated are very suitable with respect to clarity, accuracy, and usefulness of the information represented. As the topic map is encoded in XTM, the results can be displayed in graphical form by using any design tool for concept maps, such as Cmap[3] or ONTOPIA. An example of the final appearance of the topic map can be seen in Fig. 2.

V. Conclusions and Future Work

In this paper, we have presented a new method to obtain knowledge information from a set of unstructured documents in natural language, with the only help of a morphological analyzer, a lexical database, a syntactic parser, and a generic Gazetteer, and without any help of the context. The idea is to summarize the set of texts using statistics methods and NLP tools, and then to use the parser to build a topic map, and the lexical database and semantics rules to simplify it. After that, we merge the results and we generate automatically an RDF/OWL file to store the knowledge. Besides, we have also developed an evaluation process to quantify how appropriate a topic map is to describe the embedded knowledge of a set of texts, so we have been able to compare our method with others. Our main contributions are:

- Developing an algorithm to fully automatically obtain a topic map from a set of texts, and therefore a knowledge representation of them using RDF or OWL.
- Introducing a new method to build a topic map from a set of sentences using a syntactic parser.
- Proposing a new way to simplify topic maps using the semantic relationships between its elements.
- Measuring quantitatively the quality of our knowledge representation by using the summaries (an intermediate product in the generation process).

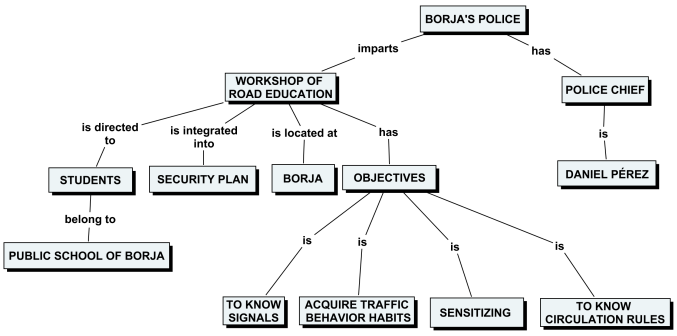

Figure 2. Example of a topic map automatically obtained from a set of texts

The main and most valuable application of this technique is to create catalogs, thesauri and ontologies that serve to

[2]http://swesum.nada.kth.se/index-eng.html

[3]http://ftp.ihmc.us

automatically categorize a repository of unstructured text-based information. The proposal has been tested in the real environment of a major Spanish newspaper and we were able to take advantage of the fact that we have thousands of texts summarized by a professional team of archivists, which has allowed us to verify accurately the results of our tests. In any case, we believe that the algorithm used is independent of the language, question that we plan to test in a short-term.

ACKNOWLEDGMENT

This research work has been supported by the CICYT project TIN2010-21387-C02-02 and DGA-FSE.

REFERENCES

[1] T. R. Gruber *et al.*, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[2] S. Pepper and G. Moore, "XML Topic Maps (XTM) 1.0 - TopicMaps. org specification," *TopicMaps. Org Authoring Group, http://www.topicmaps.org/xtm*, 2001.

[3] A. Simón, L. Ceccaroni, and A. Rosete, "Generation of OWL ontologies from concept maps in shallow domains," in *Current Topics in Artificial Intelligence*, pp. 259–267, Springer, 2007.

[4] K. M. Markham, J. J. Mintzes, and M. G. Jones, "The concept map as a research and evaluation tool: Further evidence of validity," *Journal of research in science teaching*, vol. 31, no. 1, pp. 91–101, 1994.

[5] J. Turns, C. J. Atman, and R. Adams, "Concept maps for engineering education: A cognitively motivated tool supporting varied assessment functions," *IEEE Transactions on Education*, vol. 43, no. 2, pp. 164–173, 2000.

[6] J. D. Novak and D. Musonda, "A twelve-year longitudinal study of Science concept learning," *American Educational Research Journal*, vol. 28, no. 1, pp. 117–153, 1991.

[7] K. Böhm, G. Heyer, U. Quasthoff, and C. Wolff, "Topic map generation using text mining," *Journal for Universal Computer Science (J. UCS)*, vol. 8, no. 6, pp. 623–643, 2002.

[8] L. Kásler, Z. Venczel, and L. Z. Varga, "Framework for semi automatically generating topic maps," in *Third International Workshop on Text-based Information Retrieval (TIR'06)*, vol. 205, pp. 24–30, CEUR Workshop Proceedings (CEUR-WS.org), 2006.

[9] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and development*, vol. 2, no. 2, pp. 159–165, 1958.

[10] C.-Y. Lin and E. Hovy, "Identifying topics by position," in *Fifth Conference on Applied Natural Language Processing (ANLC'97)*, pp. 283–290, Association for Computational Linguistics, 1997.

[11] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz, "Multi-document summarization by sentence extraction," in *2000 NAACL-ANLP Workshop on Automatic Summarization (NAACL-ANLP-AutoSum 2000), volume 4*, pp. 40–48, Association for Computational Linguistics, 2000.

[12] S. B. Silveira and A. Branco, "Extracting multi-document summaries with a double clustering approach," in *Natural Language Processing and Information Systems*, pp. 70–81, Springer, 2012.

[13] S. Sekine and E. Ranchhod, *Named Entities: Recognition, Classification and Use*. John Benjamins, 2009.

[14] L. L. Hill, "Core elements of digital gazetteers: placenames, categories, and footprints," in *Research and Advanced Technology for Digital Libraries*, pp. 280–290, Springer, 2000.

[15] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, 1988.

[16] D. Grune and C. J. Jacobs, "Parsing techniques-a practical guide," 1990.

[17] L. Maicher and H. F. Witschel, "Merging of distributed topic maps based on the Subject Identity Measure (SIM) approach," *Proceedings of Berliner XML tags*, vol. 4, pp. 301–307, 2004.

[18] D. L. McGuinness, F. Van Harmelen, *et al.*, "OWL web ontology language overview," *W3C recommendation*, vol. 10, no. 2004-03, p. 10, 2004.

[19] S. Pepper, "The tao of topic maps," in *Proceedings of XML Europe*, vol. 3, 2000.

[20] S. B. Silveira and A. Branco, "Using a double clustering approach to build extractive multi-document summaries," in *Text, Speech and Dialogue*, pp. 298–305, Springer, 2012.

[21] R. Navigli, "Word sense disambiguation: A survey," *ACM Computing Surveys*, vol. 41, no. 2, p. 10, 2009.

[22] H. Dalianis, *SweSum: A Text Summerizer for Swedish*. KTH, 2000.

[23] X. Carreras, I. Chao, L. Padró, and M. Padró, "FreeLing: An open-source suite of language analyzers," in *Fourth International Conference on Language Resources and Evaluation (LREC'04)*, pp. 239–242, European Language Resources Association, 2004.

[24] P. Vossen, *EuroWordNet: A multilingual database with lexical semantic networks*. Kluwer Academic Boston, 1998.

[25] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *ACL Workshop on Text Summarization Branches Out (WAS'04)*, pp. 74–81, 2004.

[26] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*, vol. 1. Cambridge University Press, 2008.