Development of a WordNet Prototype for the Macedonian Language

Martin Saveski

BSc (Hons) Computing Science
Staffordshire University

A project submitted in partial completion of the award of the degree of BSc (Hons) Computing Science from Staffordshire University franchised program at New York University Skopje

Supervised By:
Dr. Igor Trajkovski
May, 2010

Acknowledgements

First of all, I would like to thank my parents for their continued moral support and positive encouragement throughout my studies. Special thanks go to my supervisors Dr. Igor Trajkovski and Dr. Vladimir Radevski for their enthusiastic guidance and invaluable advice during the entire course of this project. Thanks also go to Dr. Katerina Zdravkova who provided me a copy of the Macedonian lexicon. To Dr. Darja Fiser and Dr. Gordana Pavlovic-Lazetic who supplied me the Slovene and Serbian WordNets, respectively. Finally, I would like to thank my friend Goce Gochevski for making comments and proofreading my thesis.

Table of contents

List of Figures

List of Tables

Chapter 1

Context and Preliminary Investigation

1.1 Introduction and Background

Natural Language Processing (NLP) is a field of computer science and linguistics whose main objective is through the use of computers to process written and spoken language for some practical, useful, purpose. The idea of giving computers the ability to understand human languages is as old as the idea of computers themselves. The benefits of realizing this idea are unlimited. However, to “understand" languages, computers must know what concepts a word or phrase stands for and know how to link those concepts together in a meaningful way. It is ironic that natural language, the symbol system that is easiest for humans to learn and use, is hardest for computers to master [1]. In order to bring computers closer to the way in which humans think and understand the natural languages, scientists have proposed a wide variety of methods to encode and manipulate with the knowledge that humans possess and use in their everyday life.

Namely, *semantic networks* are one of the most popular Artificial Intelligence concepts for knowledge representation that have been widely used in the 1970’s and 1980’s to structurally represent knowledge. Just like other networks, they consist of nodes and links. More precisely, nodes represent concepts, abstract classes whose members are grouped according to their common features and properties, and links represent relations among the concepts and are labeled to indicate the relation they represent. However, the semantics of the concepts do not residue in the network labels, but in the properties of the concepts and their relation to other concepts in the network. In the last decades the interest of building large scale semantic networks is increasing. One of the most notable is the impressive work on the CYC project [2], which developed a semantic network with more than 60.000 facts and room for more growth. Many scientists argue that the ontological representation of general and domain specific knowledge is a must to any attempt to intelligently solve the hard problems encountered in the modern computer science.

A bit different form of the traditional semantic networks was proposed by Professor George A. Miller and his colleagues from the Cognitive Science Laboratory from Princeton University. They developed the concept of a *lexical semantic network*, in which the nodes represented sets of actual English words that share a common meaning. These sets of words, called *synsets* (synonymy sets), comprise the main building blocks for representing the lexical knowledge modeled in WordNet, the first implementation of lexical semantic networks. Like in the traditional semantic networks, the semantics of the lexical nodes (the synsets) are given by the properties of the nodes (by the synonymy relation that holds between the words in the synset) and the relations to the other nodes of the network. These relations are either semantic, connect similar concepts like the relations found in the inheritance hierarchies of the traditional semantic networks, or lexical i.e. specific to lexical semantics representation domains.

The WordNet's methodology in representing the lexical knowledge is nowadays an evident trend imposed by the significant improvements in performance and by the ease of interaction displayed by the systems that have adopted this integration. Moreover, the public release of the Princeton WordNet, encoding the English language, inspired a lot of researchers around the world to develop similar knowledge representation resources for other languages. Today, there are more sixty WordNets built worldwide for more than fifty languages [3]. A lot of contributions to this have the projects whose objective is to build and connect WordNets for multiple languages (multilingual WordNets) such as: EuroWordNet, MultiWordNet, and BalkaNet (all further explained in the following sections). In addition, the public release of WordNet made it ideal tool and motivation for researchers from various fields. A plethora of applications which use WordNet has been developed. Some of them are: part of speech tagging, word sense disambiguation, text categorization, information extraction, and so on.

The main aim of this project is to develop a WordNet for the Macedonian Language and to analyze its impact on the improvement of various NLP applications.

1.2 Motivations and Objectives

As discussed in the previous section, the benefits of building a lexical semantic network (the terms lexical semantic network, lexical database and ontology are used interchangeably) for a language are enormous both for the everyday users, as a lexical resource, and for its various applications. Unfortunately, this potential has never been utilized for the Macedonian language. Except for the traditional lexical resources such as dictionaries, currently I am not aware of any attempt being made for building a large lexical resource such as WordNet ontology for the Macedonian language. This is the main motivation for this project. According to me, building a prototype of such lexical resource would be beneficial not only for the native speakers and the people who learn Macedonian, but it will give a solid surface for further research and applications.

Macedonian is the youngest language in the Eastern group of South Slavic languages. The first works that distinguished themselves as Macedonian, rather than being of other Slavic language, are evident from the end of 10th century in a form of religious codices. However, the language itself has been codified with establishment of the contemporary Republic of Macedonia in 1945 by issuing the first grammar of the Macedonian Language, edited by Blaze Koneski. The Macedonian language is official language in the Republic of Macedonia and is spoken by 1.6 million people in the country and at least that much in the Diasporas.

Although, the manual construction of the WordNet for the Macedonian language would be most accurate and precise it would require a lot of time and resources. However, the resources and time for this project are very limited, and more importantly, the objective of this project is neither the manual construction of WordNet nor producing a platform that will aid this process. Instead, the main objective of this project is by using the available lexical resources and combining several methods for automatic construction to make an attempt for building a WordNet prototype for the Macedonian language. Although, some methods may employ other WordNet implementations, the English WordNet developed by Princeton University (PWN) is selected to be the backbone for the construction of the Macedonian WordNet. Thus, the synsets of the produced WordNet will be perfectly aligned with synsets of the English WordNet. It’s obvious that one to one bijection is not possible since many concepts that are lexicalized in English are not lexicalized in Macedonian and vice versa. But, even if a lot of parts of the produced WordNet remain in English, the produced WordNet will again be usable for many WordNet applications in Macedonian, as a consequence of the WordNet’s structure.

The following list summarizes the main objectives of this study:

- To investigate and analyze the structure of the Princeton implementation of WordNet *(Model WordNet)*,
- To explore other lexical databases built using PWN as guide, their approach and methodology *(Background Experience)*,
- Based on the analysis and the knowledge gained from the research to propose a method for automatic development of a prototype lexical database for the Macedonian language *(Fundamental Study)*,
- Develop prototype WordNet for the Macedonian language *(Main Outcome)*,
- Evaluate the results and propose methods for possible improvements *(Evaluation)*,
- Analyze the impact of the Macedonian WordNet in improvement of various NLP applications, such as word sense disambiguation, text classification, text clustering, and others *(Testing and Prospective Work)*.

Since, this project does not investigate a standard software development problem, which has certain and well defined aims and requirements, the list of objectives may be a subject to change as the project progresses.

1.3 Project Output/Deliverables

The following list briefly summarizes the most important project deliverables. The list is divided on research, implementation, and documentation deliverables.

- Research Deliverables:
    - Analysis of the structure of the English WordNet implementation developed by Princeton University,
    - Analysis of the structure of other WordNet implementations,
    - Analysis of the various metods for automated construction of WordNets,
    - A method for automated construction of WordNets with emphasis on the Macedonian language,
    - Critical appraisal of the proposed method,
    - Analysis on the impact of the proposed WordNet prototype in improvement of various NLP applications.

- Implementation Deliverables:
    - English-Macedonian Machine Readable Dictionary (intermediate result),
    - A prototype of the Macedonian WordNet (in several formats),
    - Visual Web Interface of the proposed WordNet prototype (to be developed in the future).

- Documentation Deliverables:
    - Written report dissertation conformant to the Staffordshire University standards,
    - Project Plan detailing the activities carried out during the project execution,
    - Log Books detailing the discussions during the weekly meetings with the supervisor.

It is important to note that this list might be a subject to change as the project progresses. The reason for this is that many of the listed deliverables strongly depend on the progress made during the implementation phase of the project.

1.4 Tools and Methodologies

Due to the nature of the project none of the standard software development methods can be employed, but rather more agile software development approach will be taken in accordance to the project needs. During the project development the Python programming language was chosen to be used, mainly because of its intuitiveness and robustness. However, other programming languages might be employed as well, since many WordNet implementations provide interfaces thorough other programming languages. In addition the analysis and implementation of the methods automated construction of WordNets may require other NLP tools to be used.

1.5 Personal Challenges

During my high education studies, I have found the field of Artificial Intelligence as most interesting and most enjoyable to explore. However, besides the short project assignments in the two modules devoted to this field, I have never had a real opportunity to deeply research, analyze, or give any contribution on some aspect in the field of Artificial Intelligence.

As discussed in the introductory section, many times it has been concluded that the hard problems in modern computer science cannot be solved without the use of the complex ontologies. Particularly, in the field of Natural Languages Processing (NLP), the WordNet ontologies have been extensively used for complex analysis of textual corpuses and have already shown a great success. I find the development of WordNet ontology for the Macedonian language challenging, because I think that it provides a sound surface for further research and progress in the field of Macedonian language processing and Artificial Intelligence methods. It represents a great challenge for me to deeply analyze the existing methods and by combining them to propose a method for fully automated construction of WordNets. This method will be implemented and extensively tested by constructing the Macedonian WordNet prototype. However, all this to be achieved I will have to test and experiment with the existing methods and to find their advantages and disadvantages. For this reason, I will have to familiarize myself with the various NLP tools and techniques that are used by the methods. I might be required to collect large Macedonian text corpus, possibly translations of Macedonian texts (in various languages), construct parallel corpora, aggregate many machine readable dictionaries, use and manipulate WordNets for other languages, etc. All this will give me a great opportunity to gain new practical knowledge and to strengthen my academic abilities.

Lastly, I will be most satisfied if the WordNet prototype that I plan to build can be used to improve the existing methods for Macedonian text analyses or be applied in various NLP products for the Macedonian language.

1.6 Investigation Method

The method that will be used during the investigation of the project consists of finding the available resource and indentifying the ones that are relevant and helpful for the analysis. The resources will mostly consist of books, scientific publications, software tools, available databases, and text collections. In addition, the official web sites of the projects that aimed to build WordNets will be used as leading source for finding further details about the WordNet projects planned to be investigated. The Global WordNet Association, which is an organization that provides a platform for discussing, sharing and connecting WordNets for all languages in the world [4], provides many resources and links to materials about the existing WordNets. Books will be mostly used to investigate the structure of the WordNets, especially Fellbaum, C. [5] and Vossen, P. [6]. On the other hand, the applications of the WordNets and the methods and heuristics for building WordNets will be investigated by using scientific publications.

1.7 Ethical Appraisal of the Project

Since the area is general and not well standardized for any language, one of the major objectives of my work will be to make a detailed investigation of the available resources with consideration of the given ethical standards, property aspects, who generated the resources, in which area this resources are available and where. Briefly, out of enormous amount of resources available and planned to be used during the construction of the WordNet, to classify the resources according to their intellectual properties, requirements, and where needed to communicate with the authors in order to use their resources for the project. On the other side, a careful analysis and description will be provided on all methods I will use or propose during the work. Moreover, I will try to identify the ethical implications or issues that may arise during my work and I will take the actions necessary to address them. Finally, the project will contain a clear identification of the ownership of the databases, ontologies, methods, and algorithms used, including an emphasis on the new contributions that I hope to develop while working on this project.

1.8 Project Plan

The project plan breaks down the project according to the tasks that are planned to be done during the project execution. In addition, it provides a way to make sure that the project is progressing as planned. The project plan was built taking in consideration the obligations toward the other four modules that I attend during the first stage of the project. The project plan and the logbooks which contain more information about the supervisory meetings are included in the appendices A and B, respectively.

Chapter 2

Analysis

2.1 Analyses of the English WordNet

2.1.1 Foundations

Lead by the fact that if computational linguistics ever want to process natural languages as people do, there must be a store of lexical knowledge available as extensive as people have, the group of scientists from the Princeton Cognitive Science Laboratory got an idea on building a lexical database. From the previous assumption they concluded that comprehensive lexical database has to be built which would include word meaning as well as word forms to be processed by a computer. Inspired by the way in which many cognitive psychologists and computational linguists were formulating the word meanings in terms of networks, diagrams with nodes to represent word relations and semantics, the team realized that this may be a good principle of creating a lexical database.

Initially, they thought that it is most important to establish the set of nodes in the network and the semantic relations between them. Their idea was that, if they manage to establish the relations correctly, to model the real world, the patterns between the words would be enough to define the words semantics. The inclusion of glosses seemed redundant.

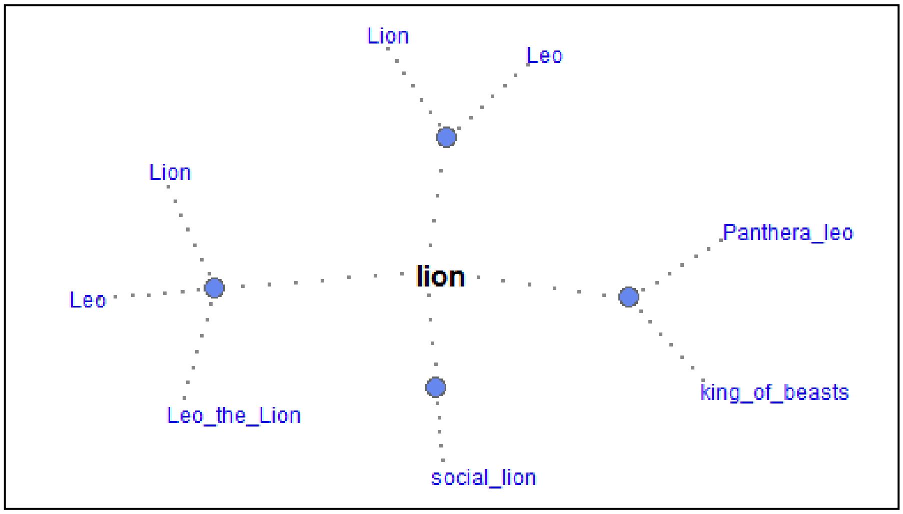
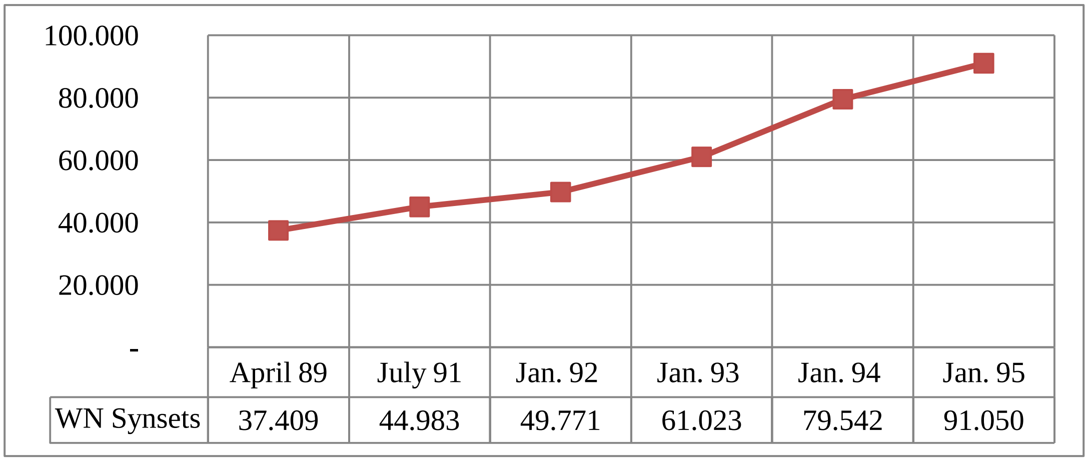


Figure 2.1: Synsets of the word “lion”

With such ambitions Professor George A. Miller, the project initiator, in 1984 when presenting his ideas to the National Institute of Education built a small network of 45 nouns on an IBM PC and he called it WordNet. With this the initial work on the WordNet project started. Afterwards, Professor George A. Miller with his wife and several undergraduate students began to work seriously on what was going to become WordNet. With the first application of WordNet, the WordFilter (automatic simplifier of text styles), they gained good directions how the project should develop in the future. As the time passed the project grew and many people had been involved to work on parts of it. The chart below depicts the growth of WordNet in terms of number of synsets (WordNet's main building block, explained in the next sections) from April 1989 to January 1995. Figure 2.1 shows the synsets in which the word *lion* is contained.



	April 89	July 91	Jan. 92	Jan. 93	Jan. 94	Jan. 95
WN Synsets	37.409	44.983	49.771	61.023	79.542	91.050

Figure 2.2: Development of PWN over the years

2.1.2 The structure of WordNet

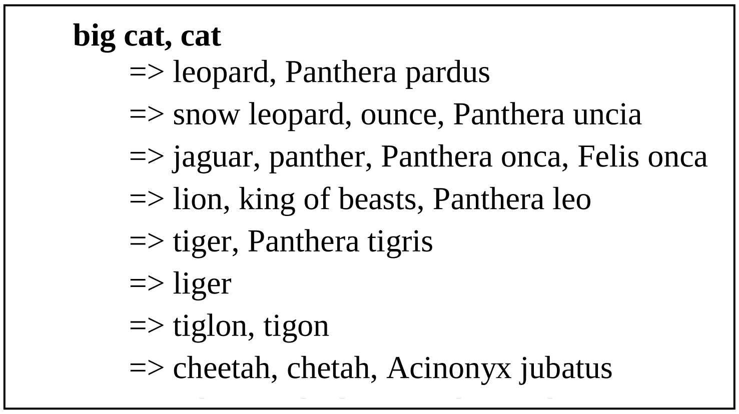
Before WordNets the lexicon of a language was represented only by dictionaries. However, the dictionaries are not meant to show the lexicon's structure of the language, but to enable the users to find information about words spelling, meaning, and use. On the other hand, thesauruses take the word's meanings as their organizing principle and are designed to help users find the "right" word when they have concept in mind.

WordNet is neither a traditional dictionary nor a thesaurus, but combines features of both types of lexical reference resources. Thus, the user of WordNet who has a given concept in mind can find, by searching one of the words expressing this concept, other words that lexicalize the same concept. Furthermore, WordNet does much more than just listing concepts in form of synsets (synonym sets), it links them by a number of relations. Unlike the thesaurus, the relations between concepts are made explicit and labeled, users are able to select a relation that will guide them from one concept to the next and choose the direction of their navigation in the conceptual space. In addition, not always the concepts are lexicalized in a language, although words depend on the existence of concepts the inverse is not the case, concepts can exist independently. Because of the way in which WordNet links concepts together, it reveals conceptual inventory that is only partially mapped onto the lexicon of English. In this respect, WordNet mainly differs from thesauruses where only lexicalized concepts are included.

The basic semantic relation in WordNet is the *synonymy*. Set of synonyms called *synsets* form the basic building block. However, the notation of synonymy used in WordNet does not entail interchangeability in all contexts. As G. A. Miller states in [5] chapter 1, "It is convenient to think of a synset as representing a lexicalized concept of English. That is to say, a lexicalized concept is represented in WordNet by the set of synonyms that can be used (in an appropriate context) to express that concept". The synonymy is a relation which holds among all types of words, but it is obvious that not all words share the same properties. For this reason WordNet divides the words in four categories: nouns, verbs, adjectives, and adverbs, each of which is treated differently.

2.1.3 Nouns in WordNet

Nouns are by far the most numerous category in WordNet. The lexical hierarchy of nouns in WordNet is achieved by the relation of *hyponymy,* which is moving from more general to specific concepts. Respectively, the inverse relation, *hypernymy* moves from more specific to more general concepts. Therefore, in WordNet moving toward more specific terms is as easy as moving toward more general terms. Consider the following example of hyponymy for one of the senses of the noun "cat":



big cat, cat
=> leopard, Panthera pardus
=> snow leopard, ounce, Panthera uncia
=> jaguar, panther, Panthera onca, Felis onca
=> lion, king of beasts, Panthera leo
=> tiger, Panthera tigris
=> liger
=> tiglon, tigon
=> cheetah, chetah, Acinonyx jubatus

This property (tree diagram, hierarchy) of WordNet uses one of the main principles in computer science, inheritance. Thus, all of the properties of the more general terms are inherited or assumed to be properties of the specific terms as well. In the example above, leopard has all the properties that big cat has plus his more specific ones. Enormous amount of information can be inherited in this manner. Although the general structure of the words is described by these relationships, it is not obvious how this knowledge is represented in a person's lexical memory, i.e. how people differentiate one concept form another. It seems reasonable to assume that the conceptual details distinguish between concepts.

Therefore, another relationship considered in WordNet is the *part-whole semantic relationship* between nouns. It is called *meronymy* from the Greek word *meros* which means part. This relation also has an inverse:

if S_m is a meronym of S_h, then S_h is said to be holonym of S_m.

The conventional phrases are "is a part of" or "has a". If W_m is a part of W_h is suitable, then W_m is meronym of W_h; similarly if W_h has a (as part of) W_m is suitable, then W_h is holonym of W_m. Figure 2.3 visually illustrates the meronymy and hypernymy relationships.

The meronymy is often compared to hyponymy since both are asymmetric and transitive (meronymy is not always), and both can relate terms hierarchically. In other words, we can say that finger is part of hand, a hand is part of an arm, an arm is part of body; i.e. parts can have parts. However, the transitivity property of this relation can imply illogical conclusions. For example, the branch is part of the tree and the tree is part of the forest doesn't imply the branch is part of the forest. To solve this, many different part of relationships has been proposed, some of which are not transitive. In WordNet there are three types of meronymy:

W_m *#p* -> W_h indicating that W_m is a component part of W_h,
W_m *#m* -> W_h indicating that W_m is a member of W_h, and
W_m *#s* -> W_h indicating that W_m is the stuff that W_h is made from.

As the psycholinguists suggest, *antonymy* is the strongest association between two words [5]. Proof for this is the word associations test. For example, if people are asked for the first word they would think of (other than the word itself) when they hear male, most will respond female; when they hear female they respond male. Thus, although antonymy is not a fundamental organizing relationship, it is present in WordNet. As G. A. Miller indicates in [5] the most interesting observation about antonymous nouns is that noun antonyms almost always have the same hypernym, often the same immediate hypernym.

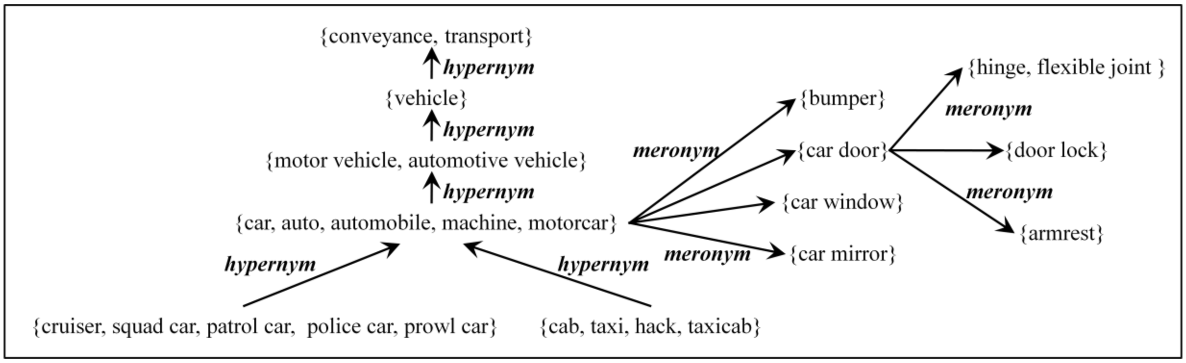


Figure 2.3: The first sense of the word car

2.1.4 Verbs in WordNet

In English there are far fewer verbs then nouns, and verbs are approximately twice as polysemous as nouns. This makes the task of structuring the verbs very difficult. Before WordNet most of the semantic networks have focused on nouns, so it was up to the team developing WordNet to establish lexical and semantic relations that will accommodate all the English verbs synsets. The verbs in WordNet are divided in two categories: verbs denoting *actions* and *events*, and verbs denoting *states*. Because, most of the verbs belong to the first category, it is further subdivided in 14 specific domain categories denoting: verbs of motion, perception, contact, communication, competition, change, cognition, consumption, creation, emotion, possession, bodily care and functions, and verbs referring to social behavior and interaction.

The main relation among verbs in WordNet is *entailment*. Entailment is referred to the relation between two verbs V_1 and V_2 that holds when the sentence “Someone V_1” logically entails the sentence “Someone V_2”, where entailment is one-sided relation. Entailment between verbs resembles meronymy between nouns. On the other hand, to model the verbs relation of hyponymy requires many kinds of semantic elaborations across different semantic domains. In WordNet these elaborations have been merged into a manner relation called *troponymy*. Troponymy between two verbs is expressed by the conventional phrase “To V_1 is to V_2 in some particular manner”. Every troponym V_1 of a more general verb V_2 also entails V_2. Therefore, troponymy is particular kind of entailment i.e. pairs that are always temporally coexistent and are related by entailment. It is important to note that the verb hierarchies (build by troponymy) tend to be more shallow and bushy then nouns.

The *antonymy* relation for verbs in WordNet expresses a complex relation aggregating several subtypes of opposition. Most stative verbs are antonymous but have no other relation that holds them together.

Finally, the *cause* relation connects two verb concepts, one causative (like show) and the other resultative (like see). Cause is another special case of entailment, thus all the properties for entailment also hold for the cause relation.

2.1.5 Adjectives in WordNet

Adjectives are words whose function is to modify nouns (e.g. large and comfortable, in large and comfortable chair). In WordNet adjectives are divided in two categories. *Descriptive adjectives* by far comprise the larger category and *relational adjectives* which are related by derivation of nouns. Descriptive adjectives are what is associated when adjectives are mentioned. Descriptive adjectives typically assign attribute value to a noun. For example, the attribute weight of the noun can be represented by the adjective heavy. WordNet contains pointers between descriptive adjectives and nouns by which appropriate attributes are lexicalized.

The semantic organization of descriptive adjectives is very different then other types of words. Instead of hyponymy, the basic semantic relation is *antonymy*. The main reason for this is that attributes tend to bipolarize. However, antonymy is a semantic relation between word forms rather than concepts (synsets). For this reason, WordNet contains antonymy pointers between words rather than synsets. One of the problems that arise in this formulation is that not all descriptive adjectives have appropriate antonyms. To solve this problem, the *similarity pointer* is introduced. The similarity pointer connects adjectives which do not have direct antonyms with semantically similar adjectives which have direct antonyms. Thus, the adjectives that lack direct antonyms have indirect antonyms through the similar adjectives. In this way, all the adjectives that point to the same adjective form a cluster which is called *satellite synset* and the adjective which has direct antonym is called *head synset*.

The second category of adjectives is the *relational adjectives*, which consists of adjectives that are semantically and morphologically related to nouns. Typically, these adjectives modify the noun and function as a classifier. For example, musical in musical instrument is related to music and dental in dental hygiene is related to tooth, they serve to identify type of instrument and hygiene respectively. In contrast with the descriptive adjective, they do not refer to a property of the nouns they modify and so do not relate to an attribute. Since, in most of the cases they do not have antonyms WordNet contains only pointers to the corresponding nouns.

2.1.6 Adverbs in WordNet

In English most adverbs are derived from adjectives by adding a suffix, most commonly -*"ly"*. In WordNet these adverbs are connected to the adjective senses by a means of a pointer meaning *"derived from"*. Because so many adjectives are polysemous and a derived adverb usually inherits the sense of the base adjective, thus particular adverbs are connected to particular adjectives senses in WordNet. The semantic organization of adverbs in WordNet is very simple. There is no hierarchy, as for nouns and verbs; nor is there a cluster structure as for adjectives. Aside from the relation between a derived adverb and adjective, there are only synonymy and sometimes antonymy relations.

2.2 EuroWordNet, Motivations and Structure

This project has been mostly inspired by the idea of creating a system that can retrieve documents from various languages given the keywords in only one language, and according to the concepts that these words describe, rather than fixed indexing and exact word matching. To achieve this, as described in [6], the aim of the *EuroWordNet* project has been the development of a *multilingual database* with WordNets for several European languages which can be used to improve the recall of the queries via semantically linked variants in any of the languages included. EuroWordNet initially started with WordNets for four languages: Dutch, Italian, and Spanish, each of which contained about 30.000 synsets. Later, the project has been expanded to include the German, French, Estonian, and Czech WordNets, which contain between 7,500 - 15,000 synsets. The project lasted three years, form 1996 to 1999, and included eight institutions where each intuition has been responsible for the construction of its national WordNet.

To reduce the project costs and to provide maximum compatibility, the WordNets have as much as possible been built from existing resources and databases with semantic information developed in various national and European projects. Moreover, the design of all WordNets has been based on the structure of the English WordNet developed by Princeton University (PWN). The notation of a synset as a main building block and the semantic relations remained in the EuroWordNet database. However, the design has been a subject to some specific changes in order to achieve maximum compatibility across languages and to maintain the relations among them.

Most of the WordNets have been developed iteratively in two major phases: *building* and *comparison* phase. The building phase included specifying the initial vocabulary for the WordNet, and encoding the language internal relations. On the other hand, the comparison phase included: loading the WordNets in the EuroWordNet database, comparing the fragments and measuring the overlap across them. After each iteration the results have been verified and the feedback has been incorporated in the next iteration. This model has allowed the WordNets to be build relatively independently while guaranteeing minimal compatibility.

To further increase the compatibility, before the beginning of the project the teams which built the WordNets defined a set of so-called *Base Concepts* that has been used as a starting point for the core of the WordNets. The WordNets have not been built to the same extend as the PWN, but rather included the generic and basic words that are related to the specific concepts and to the words which appear most frequently in the general corpora.

Further, each of the WordNets has been stored in a central lexical database and the word meaning has been linked to the meaning in the PWN, which functions as the so-called *Inter-Lingual-Index (ILI)*. The diagram below depicts the global architecture of the EuroWordNet and represents the functionality of the Inter-Lingual-Index [7].

As discussed by Vossen P. in [6], the architecture makes difference between *language specific* modules and *language independent* modules. The *language specific* modules (English, Dutch, Spanish, and Italian WordNets in the diagram) are unique and represent the internal relations between the synsets.

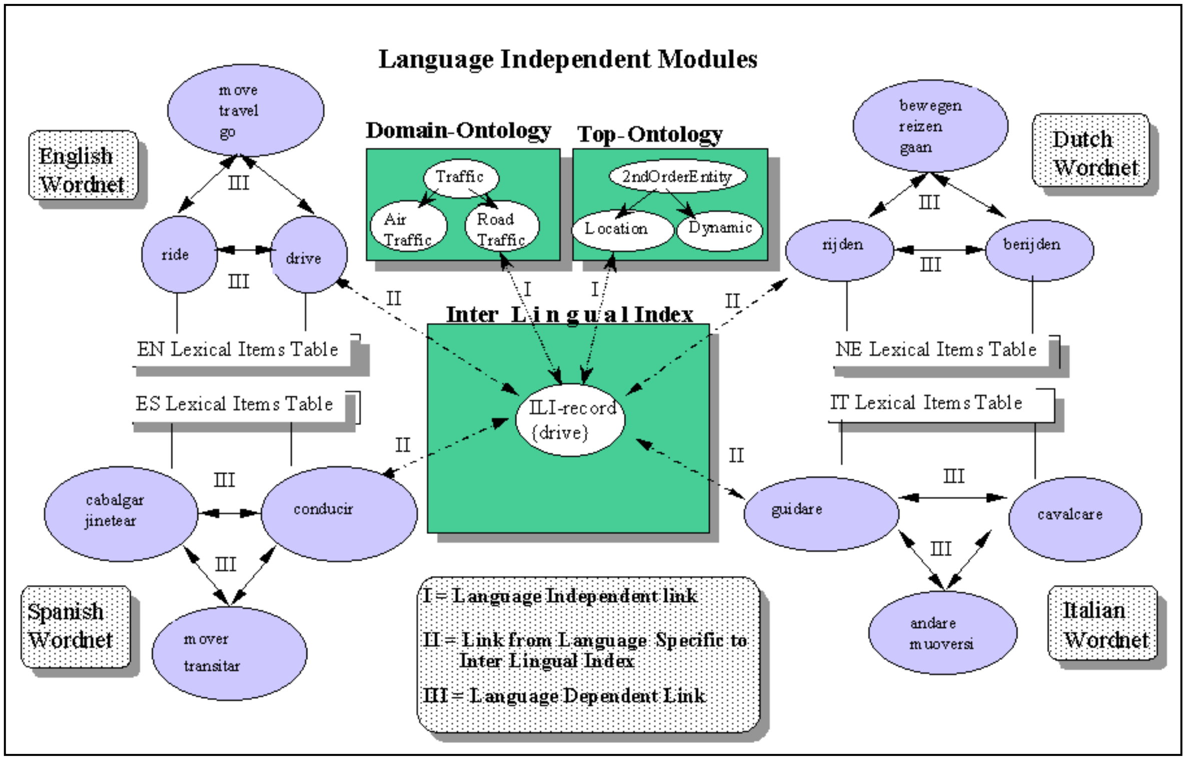


Figure 2.4: Architecture of the EuroWordNet Data Structure [7]

The *language independent* modules (Domain-Ontology, Top-ontology, and Inter Lingual Index) aim to mediate the relations between the synsets of the language specific modules. The *ILI* is consisted of *ILI-records* (mostly taken from the PWN) which are connected to the synsets of the language independent modules and possibly to one or many Top Concepts (form the Top Ontology), or domains. The *Top-Ontology* is language independent hierarchy which contains concepts that reflect important semantic distinctions such as Location, Substance, Object, etc. The *Domain-Ontology* is consisted of domain labels which structure knowledge meanings in terms of topics or scripts. The aim of the Top-Ontology and the Domain-Ontology is to provide a common hierarchy for the most important concepts in the language specific modules. By using the links within the language specific modules and between language specific and independent modules it is possible to transfer from synsets in one language to synsets in other languages representing the same concept.

A method for cross lingual text retrieval using EuroWordNet which promises successful results has been proposed in [8]. However, there are no available resources which explain the implementation of this method and confirm the expected results.

2.3 BalkaNet

As stated in [9], the main objective of the *BalkaNet* project has been building a *multilingual lexical database* consisting of WordNets for several Central and Eastern European languages. The project lasted three years, from 2001 to 2004, and developed aligned WordNets for the following Balkan languages: Bulgarian, Greek, Romanian, Serbian, Turkish, and extended the Czech WordNet previously developed in the EuroWordNet project [10].

The project closely followed the principles for building WordNets proposed in the EuroWordNet project and used the same concept of *Inter-Lingual-Index* (described in the previous section) to connect the individual WordNets. Moreover, it embedded some lessons learnt from the difficulties encountered during the construction of the EuroWordNet. Namely, changes has been made to the Inter-Lingual-Index, it has been aligned to Princeton WordNet 2.0 which improved the structure of the index while still maintaining compatibility with EuroWordNet. Also, the set of so-called Base Concepts defined in EuroWordNet has been revised and two new Base Concepts Sets (*BCS2 and BCS3*) has been defined to include concepts which are lexicalized in most of the languages. The development teams found the *XML* format as most suitable for encoding and manipulating the WordNets and agreed to use it as a common representation.

In addition to the WordNets, during the project execution various *free software tools* that enabled efficient management and exploitation of the WordNets have been developed. One example is *VisDic*, a multi-wordnet viewer, which locates the alignment problems and allows to be easily corrected [11]. As described in [10], as part of the project BalkaNet has been used for various Information Retrieval and Natural Language Processing applications. Namely, it showed successful results in Web Indexing, Word Sense Disambiguation, Question Answering, Information Extraction, summarization, Anaphora Resolution as well as for aligning and enriching WordNets.

2.4 Related Work: Automated Construction of WordNets

Due to the labor intensive and time consuming process of manual construction of WordNet many automated and semi-automated construction methods have been proposed. This section provides a short overview of the methods for automated construction found in the literature and considered as most interesting.

In [9] the methods for WordNet construction are generally classified in two models: *merge* and *expand*. The *expand model* refers to the methods for construction which closely follow the structure of a given WordNet, most commonly the PWN as a most reliable, and using various lexical recourses attempt to recreate, translate the WordNet in the target language. These methods result in WordNets compatible to PWN and consequently might be biased by it. However, they allow the exploiting of the other lexical resources linked to PWN as well as their applications. The *merge model*, on the other hand, includes methods which approach the construction by first creating an independent WordNet in the target language, not following the PWN structure, and then aligning the WordNet developed to some of the existing WordNets. As opposed to those obtained by the expand model, the WordNets produced by using the merge model tend to capture the language specific concepts more precisely. However, they are more complex, labor intensive, and require rich lexical resources.

As it was defined in the objectives of this project, the aim is to produce a WordNet prototype which will closely follow the PWN structure. For this reason, only the methods following the expand model have been considered. The reminder of this section contains a summary of the methods.

Fišer D. and Sagot B. in [12] used a *multilingual parallel corpus* to construct Slovene (SloWNet) and French (WOLF) WordNets. In a way similar to the experiment for enriching the EN-MK dictionary (will be explained in the subsequent chapters), a parallel corpus is POS tagged, lemmatized, sentence, and word aligned to produce a multilingual lexicon for five languages. Apart from Slovene and French, WordNets for the other languages has been already built and linked to PWN as part of the BalkaNet project. Next, each of the lexicon entries produced is assigned a synset id from the WordNet of the corresponding language. Finally, the intersection of the synset ids of the entries is computed and assigned as a synset id to the Slovene and French words in the lexicon entry.

Hiroyuki K. and Watanabe K. in [13] propose a method for automated construction of the Japanese WordNet by using bilingual (Japanese and English) *comparable corpora*. As opposed to the previously mentioned method, this method relies on two monolingual corpuses consisting of similar texts, but not parallel corpora. First, word associations based on occurrence and co-occurrence of the words are extracted from both corpuses. This relies on the assumption that words which occur many times closely to each other in the corpus have semantic relatedness. Once, the Japanese and English word associations are extracted they are aligned by using a bilingual JP-EN dictionary and their correlation is calculated. Next, for each synset a set of translations candidates is derived. The score of each candidate is defined as the sum of the correlations between the candidate and the associated words appearing in the gloss of the synset. Based on calculated scores, a set of Japanese words which will consist the synset translation is derived. Interestingly, the authors noted that using only the synset gloss is insufficient for finding adequate associations using this method. Therefore, the method has been extended to enlarge the set of words associated with the synset by retrieving texts using the synset gloss as a query.

Changki L. and JungYun S. in [14] for the purpose of construction of Korean WordNet, define the problem of WordNet construction quite differently than the previously discussed methods. Namely, each Korean word is mapped to a list of English translations, each of which is expanded with the WordNet synsets in which it belongs. Thus, the problem of WordNet construction is defined as finding the adequate English synset for a given Korean word. The authors propose six heuristics: maximum similarity, prior probability, sense ordering, IS-A relation, word match, and co-occurrence. Most interesting was found the *word match* heuristic which assigns a score to a given candidate synset according to the portion of overlapping words in the English dictionary definition of the Korean word and the English synset gloss and usage examples. Finally, to make the final decision the heuristics are combined by using decision tree learning, where manually mapped senses are used as a training data.

Barbu E. and Mititelu B. in [15] develop four other methods/heuristics for automated construction of the Romanian WordNet. Namely, the intersection, WordNet Domains, IS-A relation, and dictionary definitions heuristics are proposed. The last two were found very similar to the IS-A relation and word match heuristics mentioned in the previous paragraph. More attention was paid to the intersection (will be discussed in the subsequent chapters), and the WordNet Domains methods. The second makes use of the WordNet Domains project which linked the PWN synsets with a set of 200 domain labels from the *Dewey Decimal* library classification. By using a collection of domain classified documents, all Romanian words in the EN-RO dictionary are labeled with the same domain labels as in WordNet Domains. Thus, when translating a source synset only the translation candidates which match the synset domain are considered. These experiments were found very interesting since they are evaluated against the manually constructed Romanian WordNet and a formal measure of their performance is given.

Chapter 3

Design

3.1 Overview of the Automated Construction of the Macedonian WordNet Prototype

In the previous chapter the overall structure of WordNet as well as existing methods for automated WordNet construction were analyzed. Taking in consideration the analysis and research findings in the previous stages of the project, as well as the lexical resources at our disposal, in this chapter we define the steps that need to be taken during the automated construction of the Macedonian WordNet (MWN). Moreover, the main aim of this chapter is to explain in the details of the approach taken for the MWN construction, as well as the methods developed and adopted during the construction.

Figure 3.1 shows an overview of the automated construction of the Macedonian WordNet. The main resources used during the construction are the Princeton implementation of WordNet (PWN), and the English-Macedonian Machine Readable Dictionary (MRD) developed during the course of this project. PWN is used as a backbone for the WordNet construction. By the synsets, PWN provides the English words by which the concepts are lexicalized. The MRD, on the other hand, is used to find the translations of each of the words in the English synsets and to derive a set of candidate Macedonian words for the synsets in MWN. The crucial stage of the construction is the selection of words which will be members of the resulting synsets. This is achieved by experimenting with two construction methods:

1. Method based on Google Similarity Distance,
2. Intersection based method.

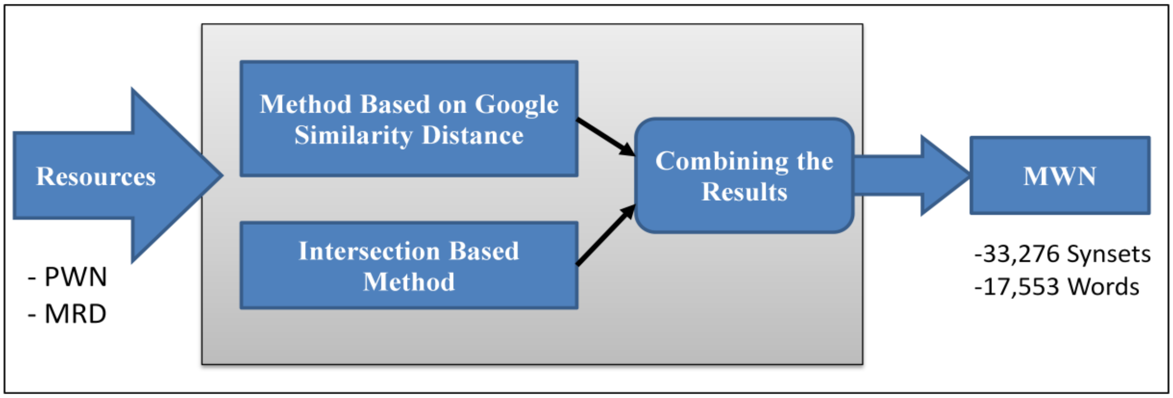


Figure 3.1: Overview of the automated construction of the Macedonian WordNet

The first method is proposed for the first time and has never been applied for automated construction of WordNets before. Both methods require that some preconditions are satisfied in order for them to be applicable. Therefore, each of the methods achieves to translate different subsets of the original WordNet. The final stage of the WordNet construction is to combine the results obtained from applying both methods into a single MWN implementation.

The remainder of this chapter discusses in more details each of the stages of WordNet construction mentioned in this section.

3.2. Method Based on Google Similarity Distance

In the following section the assumptions, formal explanation, and a walk-through example of the method proposed for automated construction of the Macedonian WordNet are explained.

3.2.1 Assumptions

In this section the assumptions on which the method proposed is based are stated. First, the method adopts the way concepts are modeled in the Princeton implementation of WordNet (PWN). Moreover, it is assumed that the conceptual space modeled by it is not depended on the language in which it is expressed (in their case English). Furthermore, we assume that the majority of the concepts exist in both languages, Macedonian and English, but only have different names. Given that the conceptual space is already represented in English by the PWN, the goal of the method is to find the corresponding concept names in the Macedonian language, by finding the proper translations of the synset members.

We are also aware of the fact that the WordNet produced will be strongly influenced by the effectiveness in which PWN conceptualizes the world. Moreover, we are aware that PWN is not a perfect lexical resource and that all of its mistakes and drawbacks will be inherited in the Macedonian WordNet produced.

3.2.2 The Method

The method mainly consists of four steps:

1. Finding candidate words,
2. Translating the synset gloss,
3. Assigning score to each of the candidate words,
4. Selection of the candidate words.

Given the above mentioned assumptions, the problem of automated construction of the Macedonian WordNet can be formulated as follows. Given a synset from PWN, the method should find a set of Macedonian words which lexicalize the concept captured by the synset.

The first step is using EN-MK machine readable dictionary (MRD) to find the translations of all words contained in the synset. These translations are called *candidate words*. Since not all English words have Macedonian translations or are not contained in the MRD, it is assumed that if more than 65% of the words contained in the synset can be translated, then the concept captured by the synset can be expressed with a subset of the candidate words. Thus, the performance of the method is strongly influenced on the size and quality of the MRD used. For this reason, as it will be elaborated in the next chapter, a lot of time and effort was spent to build a large and accurate MRD. The synsets which did not contain enough known words were skipped and retained in English.

However, not all of the candidate words reflect the concept represented by the synset. Therefore, a subset of words must be selected.

Let that the original synset contain:

$w_{1,}...,w_n$ (n English words),

and the word w_i (from this set) has *m* translations,

$cw_{1,}$ ... , cw_m, in the MRD.

Since the MRD has no means of differentiating between word senses, the set of translations of w_i, $cw_1...cw_m$, will contain the translations of all senses of the word w_i. It is a task of the method to determine which of these words, if any, correspond to the concept captured by the synset. Stated in this way the problem of translating synsets is essentially a *word sense disambiguation* (WSD) problem.

This is not very encouraging since WSD is still an open problem, but gives some pointers which may help in determining the best candidate words. Throughout the history of Artificial Intelligence many approaches and algorithms has been proposed to solve the WSD problem. Dagan I. and Itai A. in [16] stated that using the word sense dictionary definition and large text corpus, the sense in which the word occurs can be determined. In other words, the words in the dictionary definition of the word sense tend to occur in the corpus more often closely to the word in question when the word is actually in the sense defined, and less often when the word represents other senses.

Looking through the lenses of the problem of WordNet construction, this means that if the synset gloss can be translated it will give a good approximation of which of the candidate words are relevant for the synset in question.

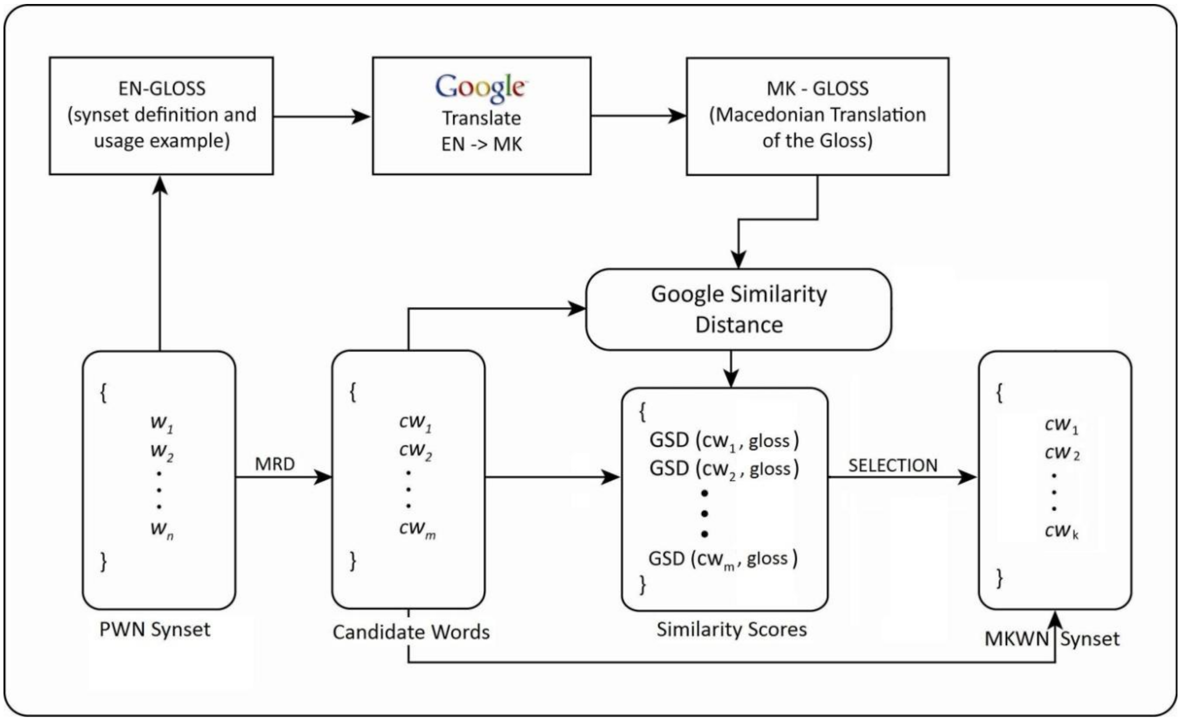

EN-GLOSS
(synset definition and usage example)
Google
Translate
EN -> MK
MK - GLOSS
(Macedonian Translation of the Gloss)
Google Similarity Distance
{ w_1 w_2 . . . w_n }
MRD
{ cw_1 cw_2 . . . cw_m }
{ GSD (cw_1, gloss) GSD (cw_2, gloss) • • • GSD (cw_m, gloss) }
SELECTION
{ cw_1 cw_2 . . . cw_k }
PWN Synset
Candidate Words
Similarity Scores
MKWN Synset

Figure 3.2: The Google Similarity Distance Method,
(n: dimension of the PWN synset, *m*: the number of candidate words,
k: dimension of the resulting synset)

Since, manual translation of the glosses is not possible, the English-to-Macedonian machine translation system available through Google on the Web can be used. Although, the Google EN-MK translation system is not extremely accurate at the time of the development of this project, its performance is sufficient enough to capture the meaning of the gloss. Form the observations, it was concluded that most common mistakes made by the translation system are inappropriate selection of the genre and case of the words. However, this does not affect the use of the gloss translation as an approximation of the candidate words and synset correlations.

The next crucial element for applying the statistical WSD technique is a large Macedonian text corpus. Although, we are aware of some small text corpuses, mostly newspaper archives available on the Web, any attempt of collecting a large, domain independent corpora is not known to exist. Using small corpora domain dependent corpora may significantly affect the performance of the method. On the other hand, collecting large textual corpus from scratch requires a lot of time and resources, which were not available for this project. Therefore, alternative method for measuring correlation between the translated gloss and the candidate words was considered.

Namely, the *Google Similarity Distance* (GSD) proposed in [17], calculates the correlation between two words/phrases based on the Google result counts returned when using the word, phrase, and both as a query. The similarity measure is further explained in the next section. Most importantly, the result of applying the GSD is a similarity score between 0 and 1 representing the relatedness of the candidate word and the translated synset gloss. The GSD is calculated for each candidate word, and the words are sorted according to their similarity.

Next, the candidate words are selected based on two criteria:

1. the words must have GSD score greater than 0.2, and
2. the words must have GSD score greater than 0.8 * the maximum GSD score among the candidates.

Finally, the words selected are included in the resulting Macedonian synset, and the other candidate words are considered as not lexicalizing the concept captured by the synset. Figure 3.2, visualizes the ideas explained in this section and the next section provides an example of construction of one synset.

3.2.3 Google Similarity Distance

Google Similarity Distance (GSD) is a *word/phrase semantic similarity distance* metric developed Rudi Cilibrasi and Paul Vitanyi proposed in [17]. The measure is based on the fact that words and phrases acquire meaning from the way they are used in the society, and from their relative semantics to other words and phrases. The World Wide Web is the largest database of human knowledge and contains context information entered by millions of independent users. The authors claim that by using a search engine, such as Google, to search this knowledge the semantic similarity of words and phrases can be automatically extracted. Moreover, they claim that the result counts of the words in question estimate the current use of the words in the society. As defined in [17], the normalized Google Similarity Distance between words/phrases *x* and *y* is calculated as:

$$GSD(x,y) = \frac{\max\{\log f(x),\, \log f(y)\} - \log f(x,y)}{\log N - \min\{\log f(x),\, \log f(y)\}}$$

where *f(x), f(y)* denote the result counts returned for *x* and *y*, respectively, and *f(x, y)* denotes the result count when both *x* and *y* are included in the query. The normalization factor *N*, can be chosen but has to be grater then the maximum result count returned. Here, the similarity distance is defined by using Google as a search engine, but is applicable with any search engine which returns aggregated result counts. The authors observed that the distance between words and phrases measured in different periods of time is almost the same. This shows that the measure is not influenced by the growth of the index of the search engine and therefore it is stable and scale invariant.

One possible drawback of the method is that is relies on the accuracy of the result counts returned. The Google index changes rapidly over time and the result counts returned are only estimated. However, linguists judge that the accuracy of the Google result counts is trustworthy enough. In [18] it is shown that web searches for rare two-word phrases correlated well with the frequency found in the traditional corpora, as well as with human judgment of whether those phrases were natural.

In [17], the GSD is applied for hierarchical clustering, classification, and language translation and showed successful results. Finally, its performance is evaluated against WordNet and resulted in mean agreement of 87%.

3.2.4 A walk-through example

To further clarify the ideas explained in the previous sections, the application of the construction method proposed is illustrated with an example. Given an English synset from PWN, the corresponding Macedonian translated synset is produced. For illustration purposes, a synset with small number of original words and candidate words was chosen. The synset *EN-06307086-N* with the following attributes was translated.

Synset	EN-06307086-N
Part of Speech (POS)	Noun
Synset definition	a defamatory or abusive word or phrase
Translation of the definition to Macedonian with Google Translate	со клевети или навредлив збор или фраза
Usage Example	sticks and stones may break my bones but names can never hurt me
Translation of the example to Macedonian with Google Translate	стапчиња и камења може да се скрши коските но имињата не може никогаш да ме повредат
Words	Name, Epithet

Table 3.1: The attributes of the synset

The table 3.1 shows a part of speech, definition, usage example, and words of the synset as well as the translations of the definition and usage examples acquired by using Google Translate. The first step is to find the synset words in the dictionary. Since, both words are included in the dictionary, the synset satisfies the rule that more than 65% of the synset words are known and can be translated. The table below shows the translations of each word. The translations with a POS different than the POS of the original synset are excluded.

Word	***Translations***
Name	презиме, углед, крсти, име, глас, наслов, слава, назив
Epithet	навреда, епитет

Table 3.2: Translations of the words in the synset

Given the translations of the words a set of candidate words is compiled. Next, by using the translated definition, the candidate word, and both, the Google Similarity Distance coefficient of each candidate word is computed. In the table below the similarity scores of the candidate words are shown. Also, the English translations of the senses the Macedonian words are included.

Macedonian Word	***English Translation of the Word Sense***	***Similarity Score***
навреда	offence, insult	0.780399
епитет	epithet, in a positive sense	0.491667
углед	reputation	0.414671
крсти	to name somebody	0.409630
назив	name, title	0.375715
презиме	last name	0.350967
наслов	title	0.350508
глас	voice	0.343176
слава	fame, famous name	0.335843
име	first name	0.334926

Table 3.3: Google Similarity Scores of the candidate words

As it can be seen from the table all candidate words are above the threshold of minimum similarity score of 0.2, but only one word is above the threshold of 0.8 * the maximum similarity score (0.624319). Therefore, only the word *навреда* is included in the translated synset. As it can be observed from the translations of the candidate words, all other words refer to other concepts. Thus, by using the Google Similarity Distance the method succeeded to select the only words which lexicalize the concept captured by the synset.

3.3 Intersection Method

The second method applied for automated construction of the Macedonian WN is the *intersection method* proposed in [15]. This method when compared with the gold standard showed most successful results during the experiments for automated construction of the Romanian WordNet. As reported by the authors, on a selected subset of synsets an error rate of only 2% has been achieved. These results are impressive, and encouraged the method to be applied for the construction of the Macedonian WordNet prototype.

The method is simple and intuitive. It exploits the fact that synonymy enforces equivalence classes on word senses. The method distinguishes two cases:

1) If the original synset contains at least one monosemous word than it is assumed that the translations of this word are unambiguous and all refer to the concept captured by the synset. In this case the synset can be translated by only including the translations of the monosemous word without considering the translations of the other words in the synset.
2) It the original synset contains more the one word, where all words are polysemous, than the translated synset are defined as the intersection of the translations of each word in the synset. In other words, only the words which appear in all translations of each word in the original synset are included in the translated synset.

If the original synset contains only one polysemous word, or the translations of the words in the synsets have intersection empty set, then the method is not applicable for translating the synset.

More formally the method can be expressed as follows. Let the original synset *EnSyn* = $\{w_1, w_2, ..., w_n\}$, where w_1, w_2, w_n are the words in the synset. Next, the translations of the words in the synset are defined as:

$T(w_1) = \{cw_{11}, cw_{12}, ..., cw_{1m}\}$

$T(w_2) = \{cw_{21}, cw_{22}, ..., cw_{1k}\}$

…

$T(w_n) = \{cw_{n1}, cw_{n2}, ..., cw_{nj}\}$

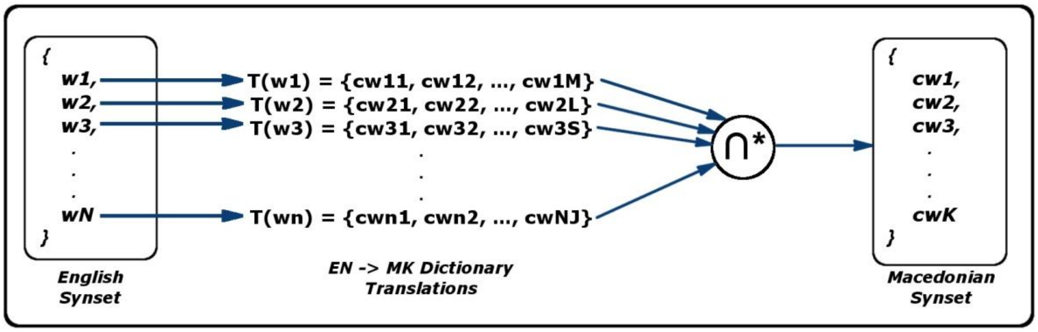

{
w1,
w2,
w3,
.
.
.
wN
}
English Synset
T(w1) = {cw11, cw12, ..., cw1M}
T(w2) = {cw21, cw22, ..., cw2L}
T(w3) = {cw31, cw32, ..., cw3S}
.
.
.
T(wn) = {cwn1, cwn2, ..., cwNJ}
EN -> MK Dictionary Translations
∩*
{
cw1,
cw2,
cw3,
.
.
.
cwK
}
Macedonian Synset

Figure 3.3: Intersection Method, Rule 2,

(N: dimension of the PWN synset, *K:* dimension of the resulting synset)

The translations synset is built as:

1) $T(w_j)$, if $\exists\ w_j \in EnSyn \mid NumSenses(w_j) = 1$

2) $T(w_1) \cap T(w_2) \dots \cap T(w_n)$, *otherwise.*

During the construction of the Macedonian WordNet using this method, the rule 2 was considered as too restrictive. Therefore, the intersection ∩ was redefined as ∩*, where a translation word is considered to belong in ∩* if it appears in more than 65% of the translations sets. Thus, by using ∩*, rule 2 is changed to be less restrictive for synsets where greater number of words from *EnSyn* are translated, while ensuring that the concept represented by the original synset is captured by the translated synset. The process of translating the synsets by using the refined rule 2 is illustrated in figure 3.3. Finally, table 3.4 summarizes the results of applying this method for construction the Macedonian WordNet Prototype.

Intersection Method	
Rule	**Synsets Translated**
1) Monosemous Word	12743
2) Intersection (∩*)	6198

Table 3.4: The results of the Intersection Method

3.4 Combining the results

The last step of the construction of the Macedonian WordNet prototype is to combine the WordNets produced by the two methods. Since, the methods rely on different rules for translating the synsets, each succeeds to translate different subsets of synsets. Figure 3.4, shows the number of synsets translated by using the Google Similarity Distance Method, the Intersection Method, and the number of synsets translated by both methods.

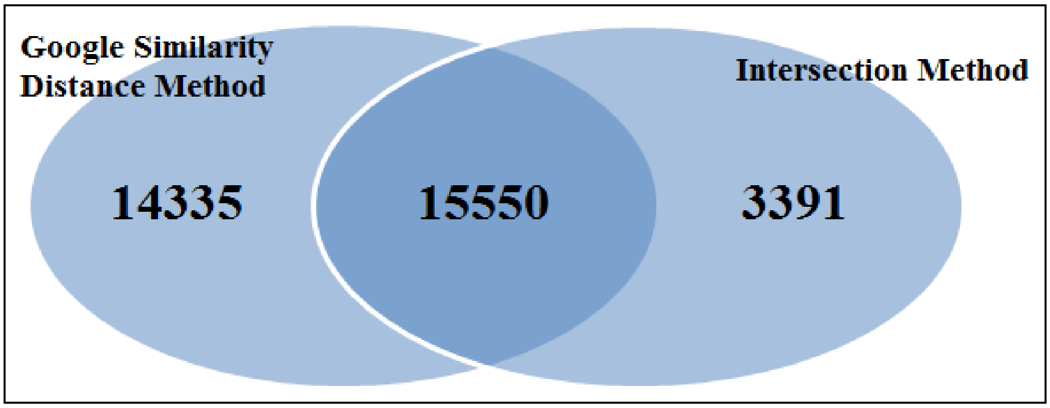
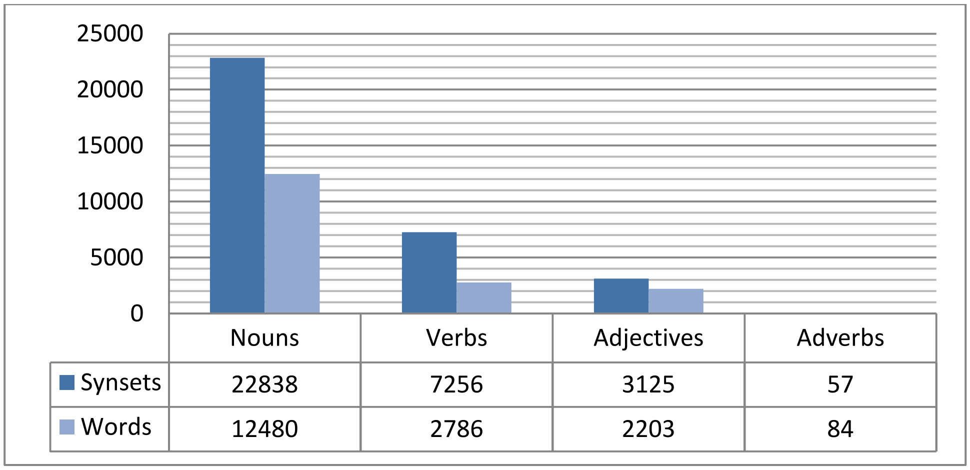


Figure 3.4: The number of synsets translated with each method

As it can be seen from the diagram, a greater number of synsets were translated using the Google Similarity Distance Method. However, by using the monosemous word translation rule, the Intersection Method succeeded to translate 3391 synsets which could not be translated by applying the Google Similarity Distance Method. Also, a large number of synsets were translated by both methods. Namely, ***45.6%*** i.e. 7088/15550 of the synsets produced by both methods contain exactly the same words. The other ***55.4%*** of the synsets in the intersection were selected by the following rules:

1) If the synset produced with the Intersection Method was translated by using the monosemous word translation rule than this synset is discarded and the synset produced with the Google Similarity Distance Method is selected.
2) If the synset produced with the Intersection method was translated by the intersection rule, than this synset is selected and the synset produced with the Google Similarity Distance Method is discarded.



	Nouns	Verbs	Adjectives	Adverbs
■ Synsets	22838	7256	3125	57
■ Words	12480	2786	2203	84

Figure 3.5: The size of the produced WordNet

The rules are based on the fact that the Google Similarity Distance Method and the intersection rule from the Intersection Method are more restrictive than the monosemous word translation rule. Finally, figure 3.5 shows the number of words and synsets translated by combining both methods, grouped by part of speech. It is important to note that all words included in the WordNet are lemmas. This is a subject of further discussion in section 4.2.4.

Chapter 4

Implementation

4.1 Overview

The main subject of the discussion in the previous chapter was the methods for automated construction of the Macedonian WordNet (MWN). However, in order to justify the concepts and clearly state the ideas, the implementation details were abstracted as much as possible. The main aim of this chapter is to explain the details behind the automated construction of the MWN.

Figure 4.1 illustrates the main steps in the development of the MWN. The development process is divided in two stages:

1. Development of a English-Macedonian Machine Readable Dictionary (MRD),
2. Implementation of the methods proposed for automated construction of the MWN.

Although, the MRD is part of the preliminary works on the WordNet construction, it is a crucial element for successful application of the methods proposed in the previous chapter. Therefore, a large amount of time and effort was spent to produce a rich and accurate MRD. The development mainly consists of: (1) acquiring and combining the existing MRDs, and (2) using parallel corpora to produce MRD.

Once, the MRD is developed the methods for automated WordNet construction are applied. The construction follows the principles defined in the previous chapter. The Princeton implementation of WordNet (PWN) as well as the MRD developed, are the main resources used during the MWN construction.

Finally, the WordNet morphological processing library (Morphy), and its corresponding implementation for the MWN are discussed.

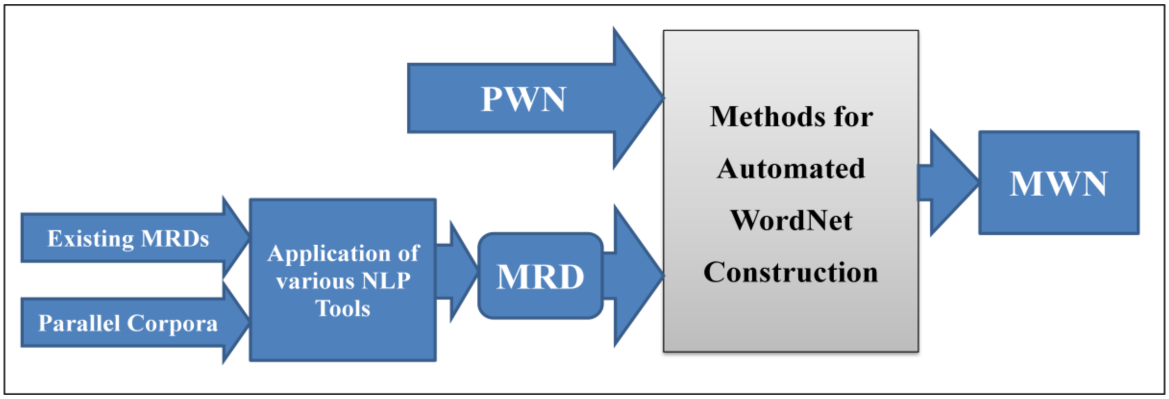

PWN
Existing MRDs
Parallel Corpora
Application of various NLP Tools
MRD
Methods for Automated WordNet Construction
MWN

Figure 4.1: The main steps in the development of the MWN

4.2 Development of a Machine Readable Dictionary

As it was discussed in the previous sections, a crucial element for successful application of the methods proposed for WordNet construction is a rich dictionary. However, it is important in the very beginning to make a distinction between the traditional dictionaries and the ones that are in machine readable form. Although, the main distinction between the two is the form in which they are stored, they usually also differ in their structure. The traditional dictionaries, for each word/phrase in the source language contain usage information, definitions, etymologies, phonetics, pronunciations, and other information in the target language. On the other hand, the *Machine Readable Dictionaries (MRD)* generally contain a list of translations, words/phrases with equivalent meaning in the target language. Depending on their purpose, some MRDs also provide word definitions and part of speech tags of the translations. The process of digitalizing traditional dictionaries to produce a MRD is long and cumbersome and is not an option for this project. It is much more convenient to make use of the existing MRD dictionaries. For this project a bilingual *English-Macedonian (EN-MK)* MRD is needed. Most of the existing EN-MK MRDs are not available on the market, but are publicly available for use on the Web. After a short investigation a list of EN-MK MRDs was compiled. The following is the list of the four most suitable resources found:

1. IDIVIDI, Macedonian web portal which contains MRDs for several languages including EN-MK MRD [19],
2. 365, similar web portal which contains MRD for several languages [20],
3. Wiktionary, Wiki based open content dictionary contains MRDs for over 350 languages including Macedonian and English,
4. Wikipedia, a MRD produced by linking keywords from the English and Macedonian version of the encyclopedia.

The number of entries is (1) and (2) was not known *a priori*, whereas (3) and (4) consisted of ~3000 and ~5000 entries, respectively. The reminder of this section explains the process of acquiring and combining these dictionaries in a single MRD.

The first step was to develop a *crawler and parser* for each of the web portals (1 and 2 in the list). This includes careful investigation of the format of the input queries, the HTML structure of the results, the HTTP request methods, and character encodings used by the portals.

Each of these factors strongly influences on the behavior of both the crawler and the parser. For example, IDIVIDI permits wildcard characters to be used in the input query. This allows all dictionary entries to be acquired by simply iterating over the three-letter combinations of all English letters. On the other hand, 365 does not allow the use of wild card characters. Thus, in order to acquire the MRD entries a list of words/phrases must be used. Fortunately, knowing the structure of the PWN database files, extracting a list of words and phrases can be easily completed. The outputs of this step are text files containing the entries of each MRD. Since, (3) and (4) are available as a text files from [21] this step was skipped.

The next step was to remove the noise in the translations while retaining information that may be useful later in the process of construction. All words, both original and translations were converted to lower case while the punctuation, long definitions, and abbreviations were removed. On the other hand, some translations contained *part of speech (POS) tags* of the translated words and/or short definitions in brackets. This information was attached to each of the translated words as its additional attribute. Figure 4.2 depicts the number of entries in each of the MRDs after this step.

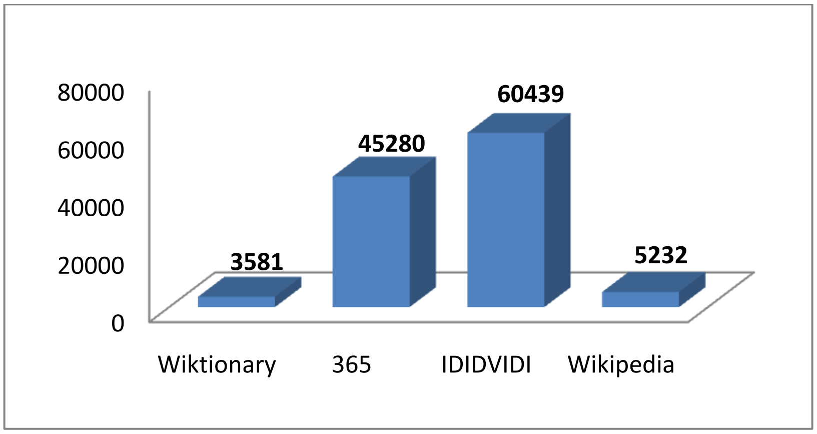


Figure 4.2: The number of entries in the MRDs

The last step was to combine all four MRDs to produce a single EN-MK MRD. Thus, for each English word/phrase there is a list of Macedonian translations. This list was produced by concatenating the lists from each of the individual dictionaries. When duplicates were encountered, only the more informative (containing more attributes) were retained. Lastly, the MRD was stored in Python dictionary, which is Python implementation of a hash table, to allow fast (constant time) access to the MRD entries.

doughy=тестен{adj},блед{adj}{лице},недопечен{adj}{леб},тестест{adj},

doup=реновира{v}{куќа,соба},поправа{v},завиткува{v}{пакет},закопчува{v},ремонтира{v}{возило},

dour=тврд{adj,adv},строг,надурен,непријатен{adj,adv},студен{adj,adv},намуртен{adj,adv},

dourly=тврд{adj,adv},строго,мрачно,непријатен{adj,adv},студен{adj,adv},намуртен{adj,adv},

douse=полева,наквасува{v},угаснува,гаси,нурнува,полива{v},намокрува,натопува,

dove=грлица{n},гулаб,мирољубив,политичар,јагне,кроток човек,

Figure 4.3: Sample output of the MRD produced

Figure 4.3 visually shows the structure of the dictionary, where the bold words on the left are English words and the coma separated list on the right contains the corresponding Macedonian translations. The words in the curly brackets next to some Macedonian words are the POS tags and additional information found in the source MRDs.

4.2.2 Using Parallel Corpora for Creating MRD

4.2.2.1 Background

Since, the MRD produced by combining existing MRDs did not fully satisfy the expectations, in terms of size, other approaches and techniques for building MRDs were considered. Although, the techniques for enriching the MRD explained in this section required a lot of time and effort, it was considered that they will not only have influence on the final results of the project, but are also fascinating and beneficial as an academic exercise.

Many *Natural Language Processing (NLP)* algorithms make extensive use of parallel corpora to achieve certain results. This is especially the case in *Statistical Machine Translation (SMT)*, which is currently one of the most popular topics is the field of NLP. More importantly for this project, by processing *parallel corpora* with various SMT tools rich MRDs can be produced. Such techniques have already been successfully applied for construction of English-Greek [22] and English-Chinese [23] bilingual MRDs. Both experiments made use of the *Uplug* system [25]. Uplug origins from a project in Uppsala University and provides collection of tools for linguistic corpus processing, word alignment and term extraction from parallel corpora. The system has been designed by Jörg Tiedemann in order to develop, evaluate, and apply approaches to generation of translation data from bilingual text [26]. Most importantly, this system is a modular-based platform which means that each component can be extended or modified without affecting the system pipeline.

4.2.2.2 Small Scale Experiment

In order to test whether this method was applicable for producing EN-MK MRD, a small scale experiment was conducted. For the purpose of the experiment the *KDE4* [27] parallel corpus has been used. This corpus is part of *OPUS (Open Source Parallel Corpus)* [27] collected from the localization files of KDE version 4, which is an open source software containing a wide variety of applications for communication, work, education and entertainment. The whole corpus contained 399,597 EN-MK tokens i.e. 71,046 EN-MK sentences, where all localization files were tokenized, sentence aligned, and stored in xml files in a format suitable for word alignment with Uplug. After the execution of the advanced word alignment module of Uplug a list of 62,565 EN-MK word alignments was produced. However, many entries contained noise and incorrect translations. Since, manual evaluation of the results was not possible, radical filtering was applied to retain only the meaningful translations. Thus, all word alignments which occurred less than three times or contained punctuation or numbers were removed, resulting in a MRD with 5,228 entries. It was also notable that most of the entries were domain specific and were not present in the previously developed MRD. These results were satisfying and encouraged further experiments with larger parallel corpus to be done.

4.2.2.2 Large Scale Experiment

For the purpose of the second experiment the data produced by South European Times (SETimes) news website was used. This website publishes daily news for all countries in south-eastern Europe and Turkey. Most importantly, the content of the web site is available in ten languages including English and Macedonian. However, unlike KDE4 this corpus was not available in any preprocessed form so a lot of preliminary work had to be done to transform it in a form suitable for applying the Uplug modules. The whole process is depicted in figure 4.4 and further explained in the remainder of this section.

CRAWLING AND PARSING: Since the only source was the official website of SETimes, the first step was to develop simple crawler and parser. The purpose of the crawler was to collect the URL for each article and to download each article in both languages. Afterwards, the parser was responsible for extracting only the articles from the HTML code and removing all unnecessary characters. Finally the articles were stored in two text files, one for each language, where one line represented one article. The content of these files was manually verified to ensure that the article in the Nth line in the first file corresponds to the translated article in the second. The articles which were missing in one language were removed from both files.

SENTENCE SEGMENTATION: The next step was to segment each article in sentences. Although, Uplug includes module for sentence segmentation, this module relies on simple rules and did not produce satisfactory results. Instead, *Punkt* was considered [24]. Punkt is a computer program which implements a language-independent unsupervised algorithm for sentence boundary detection. Understood intuitively, it is based on the assumption that a large number of ambiguities in the determination of sentence boundaries can be eliminated once abbreviations have been identified [24]. Punkt is open source, available through the Python NLTK (Natural Language Toolkit) [29] and could be easily applied to the collected corpora. To further facilitate the process of sentence segmentation, all articles that included paragraph HTML tags were first segmented on paragraphs and then sentence segmented. After this step it could be concluded that the whole corpus contains 28,980 articles i.e. 294,693 sentences.

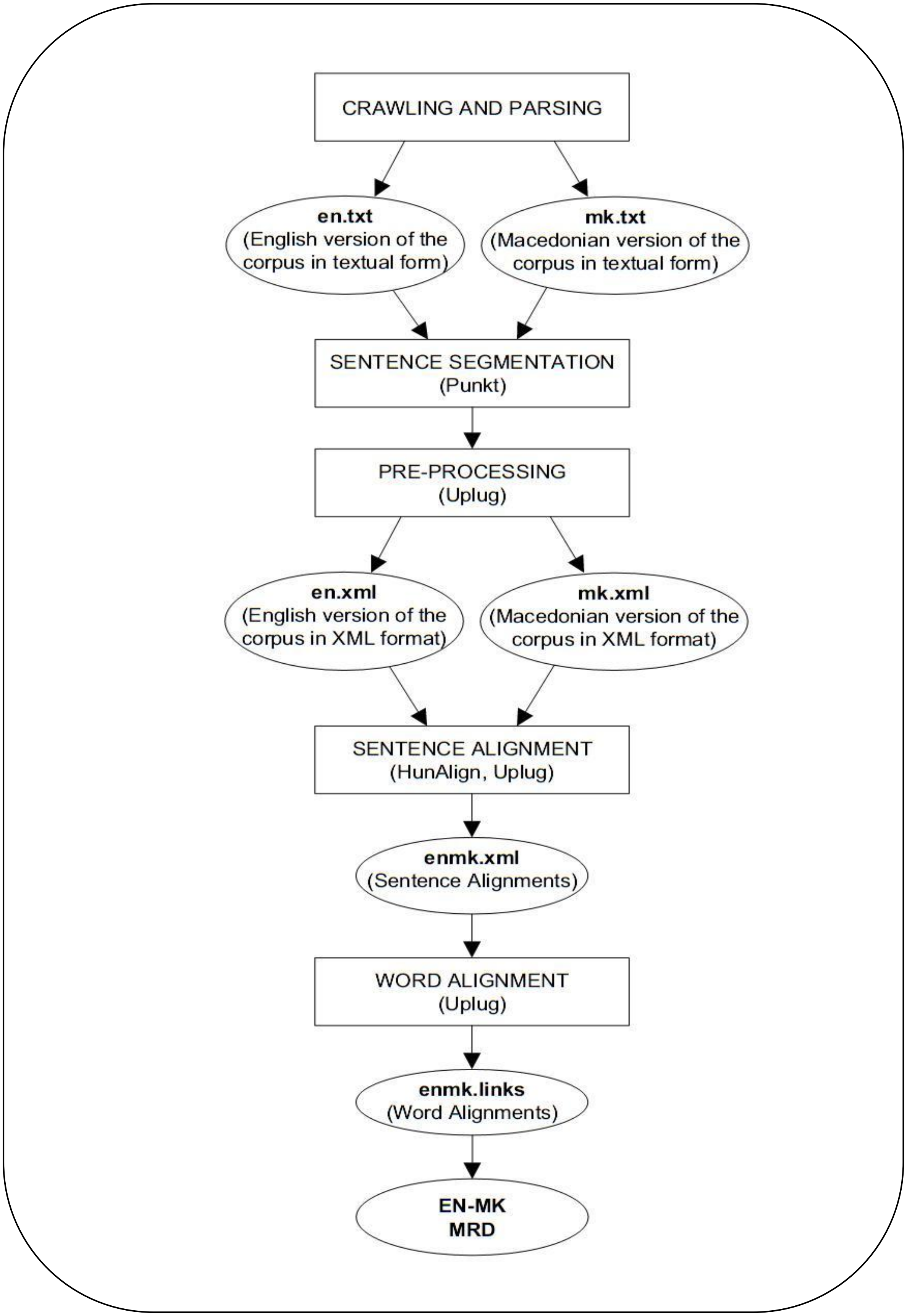


Figure 4.4: The process of producing MRD form parallel corpora

PRE-PROCESSING. Once the corpus was sentence segmented the Uplug pre-processing module was applied to allow the corpus to be further processed with other Uplug modules. The pre-processing module *tokenizes* the text and converts the text files in XML format by using basic markup for each paragraph, sentence, and word. Figure 4.5 shows a sample output from XML files produced in this step.

```
<?xml version="1.0" encoding="utf-8"?>
<text>
    <p id="1">
      <s id="s1.1">
        <w id="w1.1.1">tirana</w>
        <w id="w1.1.2">mayor</w>
        <w id="w1.1.3">to</w>
        <w id="w1.1.4">be</w>
        <w id="w1.1.5">awarded</w>
        <w id="w1.1.6">un</w>
        <w id="w1.1.7">prize</w>
        <w id="w1.1.8">this</w>
        <w id="w1.1.9">week</w>
       </s>
       ….
```

```
 <?xml version="1.0" encoding="utf-8"?>
 <text>
     <p id="1">
       <s id="s1.1">
         <w id="w1.1.1">на</w>
         <w
id="w1.1.2">градоначалникот</w>
         <w id="w1.1.3">на</w>
         <w id="w1.1.4">тирана</w>
         <w id="w1.1.5">ќе</w>
         <w id="w1.1.6">му</w>
         <w id="w1.1.7">биде</w>
         <w id="w1.1.8">доделена</w>
         <w id="w1.1.9">награда</w>
         <w id="w1.1.10">на</w>
         <w id="w1.1.11">он</w>
         <w id="w1.1.12">оваа</w>
         <w id="w1.1.13">недела</w>
       </s>
       …
```

Figure 4.5: Sample output of the Uplug pre-processing module

SENTENCE ALIGNMENT. Next, the sentence alignment module was applied. The purpose of this module is to link all sentences in one file to corresponding translation sentences in the other. Uplug contains several sentence alignment modules. After experimenting with each, it was concluded that the module which uses *HunAlign* [28] showed most satisfying results. HunAlign is language independent module which aligns sentences in bilingual texts by combining the so-called length-based and dictionary-based approaches. In the first pass of the corpus HunAlign uses the sentence-length information to build a dictionary, while in the second pass it uses the produced dictionary to realign the sentences. Furthermore, HunAlign includes one-to-many and many-to-one alignments, which allows the mistakes made in the sentence segmentation phase to be corrected with proper sentence alignment. The result of this step is a XML file containing the sentence links and the alignment certainty of each link. Sample output is shown in figure 4.6.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE cesAlign PUBLIC "-//CES//DTD XML cesAlign//EN" "">
<cesAlign toDoc="setimes/mk.xml" version="1.0" fromDoc="setimes/en.xml">
    <linkGrp targType="s" toDoc="setimes/mk.xml" fromDoc="setimes/en.xml">
        <link certainty="3.64407" xtargets="s1.1;s1.1" id="SL2" />
        <link certainty="3.374068" xtargets="s1.2;s1.2" id="SL3" />
        <link certainty="1.819944" xtargets="s1.3;s1.3" id="SL4" />
        <link certainty="4.003576" xtargets="s1.4;s1.4" id="SL5" />
        <link certainty="11.63679" xtargets="s1.5;s1.5" id="SL6" />
        . . .
```

Figure 4.6: Sample output of the Uplug (HunAlign) sentence alignment module

WORD ALIGNMENT. In the final step the word alignment module was applied to the corpus. Word alignment refers to linking corresponding words and phrases in the aligned sentences. For this purpose Uplug has three different modules: basic, tagged, and advanced. Since POS tagger for the Macedonian language was not available, to achieve best results the advanced word alignment module was used. This module includes many sub-modules which run in the following order:

1. **Basic Clues**: computes basic alignment clues using association measures,
2. **Giza-word-refined**: runs GIZA++ in both alignment directions and converts the lexical probabilities to the clue aligner format,
3. **Dynamic Clues**: learns clues from the "refined" combination of both Viterbi alignments,
4. **Gizaclue-word-prefix**: takes only the three initial characters of each token and runs GIZA++ in both directions and converts probabilities to clues,
5. **Link**: clue alignment using basic clues, GIZA++ clues, and learned clues,
6. **Dynamic Clues**: learns clues from previously aligned data,
7. **Link**: clue alignment using all clues (basic, giza, learned),
8. The last three steps are repeated 3 times. [26]

Clue alignment refers to incorporating several knowledge resources (clues) in the process of word alignment. This module is the result of extensive research and experiments conducted in [26].

The output of this step is an *XCES* XML file [30] which includes the word links and the certainty of each alignment. Figure 4.7, shows sample output of this file, where each word link element has a certainty, lexical pair, and xtragets (link word ids) attributes.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE cesAlign PUBLIC "-//CES//DTD XML cesAlign//EN" "">

<cesAlign version="1.0">
  <linkList>
    <linkGrp targType="s" toDoc="setimes/mk.xml" fromDoc="setimes/en.xml">
      <link certainty="3.64407" xtargets="s1.1;s1.1" id="SL2">
        <wordLink certainty="0.04366786" lexPair="week;недела" xtargets="w1.1.9;w1.1.13" />
        <wordLink certainty="0.02486187" lexPair="prize;награда" xtargets="w1.1.7;w1.1.9" />
        <wordLink certainty="0.03209486" lexPair="mayor;градоначалникот" xtargets="w1.1.2;w1.1.2" />
        <wordLink certainty="0.04466403" lexPair="tirana;тирана" xtargets="w1.1.1;w1.1.4" />
        <wordLink certainty="0.01992023" lexPair="be;ќе биде" xtargets="w1.1.4;w1.1.5+w1.1.7" />
        <wordLink certainty="0.02397234" lexPair="this;оваа" xtargets="w1.1.8;w1.1.12" />
        <wordLink certainty="0.02113098" lexPair="to;на на на" xtargets="w1.1.3;w1.1.1+w1.1.3+w1.1.10" />
        <wordLink certainty="0.04741557" lexPair="un;он" xtargets="w1.1.6;w1.1.11" />
    </link>
    …
```

Figure 4.7: Sample output of the Uplug advanced word alignment module

To produce more readable output the *xces-to-text* Uplug module was applied. The result is text file containing all word alignments and their frequency of occurrence. Figure 4.8, shows the corresponding output. As expected, the conjunctions occur most frequently.

44352	and	и	11401	also	исто така
24692	the	на	11209	that	дека
24538	in	во	10430	kosovo	косово
22182	with	со	9378	turkey	турција
21006	eu	еу	9352	the	на
14615	is	е	8833	as	како
13927	will	ќе	8572	was	беше
13091	on	ти	8425	not	не
12984	he	тој	7918	and	и
12950	in	во	7774	un	он
12605	serbia	србија	7711	macedonia	македонија
11708	bih	бих	7425	country	земјата

Figure 4.8: Sample output of the xces-to-text Uplug module

4.2.3 Morphy

Many dictionaries present the word translations in uninflected form, without providing a list of inflectional (derived) forms of the word. In a traditional (printed) dictionary, this causes little trouble. In English there are few exceptions, so morphologically related words generally have similar spelling and can be easily picked by the reader. In electronic dictionaries, on the other hand, when an inflected form of the word is requested it is most likely that the response will be negative. In other words, the users are required to know the base form of the word they want to look up. In order to overcome this issue, both for users and computer programs which process text, WordNet includes Morphy. Morphy is a library of *morphological processing* functions which incorporates some intelligence about the English morphology and is able to handle a wide range of morphological transformations. Morphy uses two processes to convert a word into a form that is most likely to be found in WordNet: *rules of detachment* and *exception lists*. Appling the rules of detachment involves checking lists of inflectional endings based on the syntactic category, and substituting the word suffix according to the rule matched. Figure 4.9, shows the detachment rules for verbs, similar rules exist for nouns and adjectives.

VERBS			
Suffix	**Ending**	**Suffix**	**Ending**
-“s”	-“”	-“ed”	-“e”
-“ies”	-“y”	-“ed”	-“”
-“es”	-“e”	-“ing”	-“e”
-“es”	-“”	-“ing”	-“”

Figure 4.9: Rules of detachment for verbs

These rules work as desired only with regular inflections. For this reason, for each syntactic category there is also an exception list which consists of all inflected forms followed by one or more base forms. Thus, Morphy always checks the list of exceptions before using any of the detachment rules. In a similar manner Morphy handles phrases and compounds. Noun phrases are transformed by transforming each word, whereas verb phrases are more difficult to process and require more analysis. Lastly, Morphy handles hyphenation. Since it is often a subjective decision whether a word is hyphenated or joined as one word, Morphy splits the string into “words” using both space and hyphens as delimiters.

4.2.4 Morphy for the Macedonian WN prototype

The development of a tool equivalent to Morphy for the Macedonian language would be beneficial both during the construction and use of the Macedonian WordNet. Since, the MRD used in the construction is produced by combining many MRDs and alignment of parallel corpora; it is very likely that most of its entries are in inflected (derived) word form. If such word forms are included in the WordNet, then the word space covered by it will be significantly reduced. For example, one noun in Macedonian can have more than twenty inflected forms. If the WordNet does not possess any morphological information about the words, then it can index only exact matches of a given string and is much less likely that a given query word will be found. On the other hand, if during the construction all words are *lemmatized* and also the query word is lemmatized before searching, than it is much more likely that the query word will be found. Interestingly, by using this approach, even words which did not appear during the construction i.e. appeared in other inflected form of the word lemma can be indexed.

However, Macedonian is *inflection rich language* and deriving detachment rules similar to the ones described in the previous section is extremely difficult task. Fortunately, previous research studies on the morphosyntactic structure of the Macedonian language have been conducted in the Institute of Informatics at the Faculty of Natural Sciences and Mathematics in Skopje. Namely, by using list of word lemmas and rules for inflection a lexicon of over 1.300.000 word forms has been produced in [31]. This lexicon was developed as part of the Multext-East project and is not publicly available, but it was kindly provided to us for research purposes. As shown in figure 4.10, the lexicon is stored in text file where each line contains the word form, lemma, and *morphosyntactic description (MSD)* of the word form.

Word Form	**Lemma**	**MSD**
асистентски**те**	асистентски	Aop-p-y
асистентски**ве**	асистентски	Aop-p-p
асистентски**не**	асистентски	Aop-p-d
асистенција	асистенција	Ncfsnn
асистенција**та**	асистенција	Ncfsny
асистенција**ва**	асистенција	Ncfsnp
асистенција**на**	асистенција	Ncfsnd
асистенцијо	асистенција	Ncfsvn
асистенции	асистенција	Ncfpnn
асистенции**те**	асистенција	Ncfpny
асистенции**ве**	асистенција	Ncfpnp
асистенции**не**	асистенција	Ncfpnd
асистир**ам**	асистира	Vmip1s
асистир**аш**	асистира	Vmip2s
асистир**а**	асистира	Vmip3s
асистир**аме**	асистира	Vmip1p
асистир**ате**	асистира	Vmip2p

Figure 4.10: Sample of the Multext-East lexicon

The MSD consists of several letters each of which has specified meaning explaining the word form. However, only the word form, lemma, and POS tag (the first letter of the MSD) are of interest. Thus, all functionalities that Morphy has to provide include only look ups in the lexicon. All the entries in the MRD present in the lexicon were lemmatized and POS tagged. Furthermore, to improve the process of retrieving lexicon entries while searching the WordNet, all entries whose lemmas were not included in the dictionary were removed. A summary of the MRD and lexicon sizes is shown in Table 4.1.

Resource	Number of Entries
Lexicon	27,981
MRD	1,323,718

Table 4.1: MRD and lexicon statistics

4.2.5 Implementation of the Methods for construction of MWN

The design chapter discussed the main concepts behind the methods considered for automated construction of the Macedonian WordNet (MWN). This section builds on the previously discussed concepts and elaborates the implementation details. Figure 4.11 shows the main stages in the construction of the MWN. Namely, the process consists of: parsing the Princeton WordNet database files, applying the methods for automated WN construction, and finally generating XML and WN database files. Each of these stages is further explained in the remainder of this section.

PARSING: The first step before applying the methods was to develop a parser for the PWN database files. This allows the WordNet to be read in Python data structures and to facilitate the manipulation of the synsets further in the process of construction. Although PWN is accompanied with extensive documentation, the database files follow a very specific format and their structure had to be carefully studied before the development of the parser.

GOOGLE SIMILARITY DISTANCE METHOD: As it can be concluded from the previous discussions the Google Similarity Distance Method makes extensive use of the Google services, both for searching and machine translation. The main reason for this is the easy way in which its services can be exploited. Through the *Google AJAX* APIs (Application programming interface), Google provides easy access to its services. The *Google AJAX Search API* provides an interface to the Google search service [32]. The number and frequency of the queries is not limited.

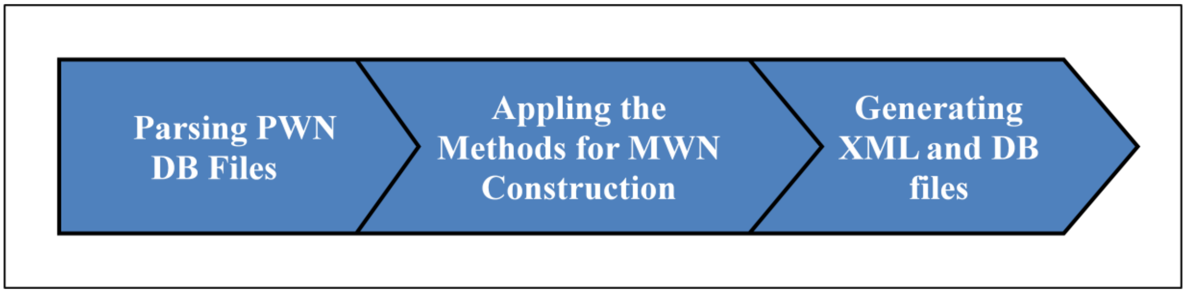

Parsing PWN DB Files
Appling the Methods for MWN Construction
Generating XML and DB files

Figure 4.11: MWN Construction Pipeline

Similarly, the *Google AJAX Language API* provides an interface to the Google translation system. This API, also does not pose any constrains in terms of the number of requests or the size of the text that can be translated during one period of time. Although, the services are mainly devoted for web development, as the name suggests (AJAX stands for Asynchronous JavaScript), the services can be easily adopted for any application. The communication in both services is through the *JavaScript Object Notation* (JSON), a lightweight data-interchange format. Python by default provides an extensive library for serialization and de-serialization of JSON objects, which once again proves it as best choice. Besides of Google other service providers such as Yahoo and Bing, the second and third most popular at the time of development of the project, were considered. However, they pose a lot of limitations in terms of the number and frequency in which queries are allowed to be sent. Moreover, Google is well-known for the large size of its index. This was visible in the preliminary experiment done before proposing the method. Google returned significantly larger result counts compared to Yahoo and Bing. For the translation system, Google was chosen since no other English-Macedonian publicly available machine translation system was known to exist. As discussed in the previous chapter, although the Google translation system is not highly effective, at the time of the development of the project, its performance is sufficient for the purpose of this project.

Since the process of translating the glosses and acquiring the result counts required a lot of time they were done in sequential manner. All glosses flagged for translation were written in file, then translated and saved. Afterwards, all queries were generated and sent to the search service and the result counts were saved. Finally, using the result counts the subsequent steps in the method were applied.

INTERSECTION METHOD: The Intersection Method does not require communication with any external resources and therefore was much simpler to implement. The main resource on which the method relies on is the MRD. More details about the dictionary and its development are included in the previous sections. For simpler and faster manipulation of the candidate words sets an extensive use of the Python *SET* data structure was made.

GENRATING XML AND DB FILES: The final step of the implementation was the development of a script which from the WordNet synsets produced will generate the corresponding WordNet database and XML (Extensible Markup Language) files. The database files generated follow the same format and structure of the PWN database files, and therefore can be used by any application which is able to use the PWN database. The XML files follow the DTD (Document Type Definition) structure proposed by the Global WordNet association defined for encoding WordNets in a unified way [4].

Chapter 5

Testing

5.1 Background

Most common practice for evaluation of the quality of the automatically built WordNets is manual verification. Subsets of the WordNet are selected and manually verified by lexicographers. Such examples are the evaluation of the Slovene [12] and Korean [14] WordNets. In the cases where manually developed WordNet already exists for the language in question, this WordNet may be considered as a golden standard and can be used to evaluate the automatically constructed synsets. Examples are the evaluation of the methods for automated construction of the French [12] and Romanian [15] WordNets.

Although, the manual verification of the synsets developed during the automatic construction of the Macedonian WordNet prototype would be the most accurate and objective evaluation, it would require a lot of time and resources. Taking into account the time and resources devoted for the development of this project, the manual evaluation was not possible. Also, as previously mentioned, this is a first attempt to build a WordNet for the Macedonian language, and therefore there is no golden standard against which the WordNet synsets developed can be evaluated.

But more importantly, the initial objective of this project was not to develop a WordNet which will be a perfect lexical resource, but rather to develop a resource which will give us the opportunity to include semantics in the already developed techniques for *Machine Learning* (ML) and *Natural Language Processing* (NLP). Therefore, it was considered that it is much more suitable to evaluate the WordNet developed by its performance in a particular NLP/ML application and by the possible improvements that its usage may allow.

Namely, we were interested in how the use of the WordNet developed will influence the performance of the text classification algorithms. As mentioned in the previous chapters, this is only one of the plethora of applications of WordNet. It was considered mainly because the performance of the classification algorithms can be measured unambiguously and compared easily. The remainder of this chapter explains: how the WordNet can be applied for finding semantic similarity between texts, the classification algorithm used for the experiment, as well as the presentation and evaluation of the results of the experiment.

5.2 Synset Similarity Metrics

The first step towards defining a method for measuring the semantic similarity between two text documents using WordNet is to define how the distance between two WordNet synsets can be measured. In this section two synset similarity metrics, *Leacock-Chodorow* and *Wu-Palmer,* are defined. Besides the metrics discussed in this section several other metrics have been proposed by: Resnik, Jiang–Conrath, Lin, and others [33]. These metrics in addition to WordNet rely on text corpora to measure the semantic similarity between synsets, and therefore were not considered for the purpose of this experiment. The interested reader may find further information and comparison between the metrics in [33] and [34].

5.2.1 Leacock and Chodorow Synset Distance

In the course of their attempt to minimize the problem of sparseness of training data for their statistical local-context word-sense classifier [35], Leacock and Chordorow explored the possibility of using WordNet to measure the semantic similarity between word senses. Namely, they have proposed the following formula for computing the semantic distance between two synsets in WordNet:

$$sim_{LCH}(s_1, s_2) = -\log \frac{len\ (s_1, s_2)}{2 * D}$$

where *len*(s_1, s_2) refers to the number of nodes in the path from s_1 to s_2, and D refers to the maximum depth in the WordNet hierarchy. It is important to note that the path is measured by the number of nodes, rather than the number of links. Therefore, the length between two sister nodes is 3 (one for each sister and one for the parent), and the length of the path between synonyms i.e. the same synset is 1. In order to ensure the existence of path between any two synsets, they have also introduced a global root node which connects all top nodes in the WordNet hierarchy.

5.2.2 Wu and Palmer Conceptual Similarity

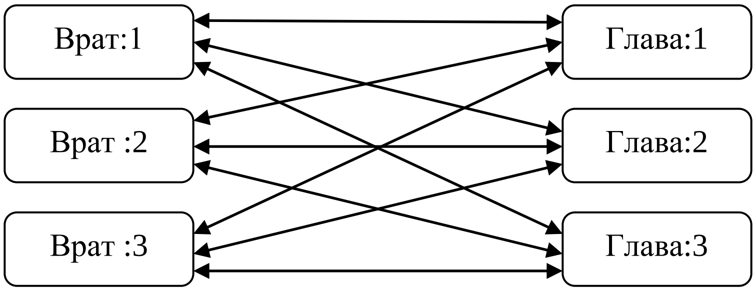
In their paper [36] focusing on the semantic representation of verbs in computer systems and its impact on lexical selection problems in machine translation, Wu and Palmer devoted a section to introduce a conceptual similarity metric between two concepts. Although, initially the metric has not been proposed for WordNet, later in the literature the same principle has been adopted for measuring the similarity between the synsets in WordNet. The metric namely measures the depth of the two given synsets in the WordNet hierarchy, the depth of the Least Common Subsumer (LCS), and combines these measures into a similarity score as follows:

$$sim_{WUP}(s_1, s_2) = \frac{2 * depth(LCS)}{depth(s_1) + depth(s_2)}$$

where *LCS* is the most specific synset which is ancestor to both synsets. Therefore, the similarity score is always in the range of (0,1]. The similarity score can never be 0, since the depth of the LCS is always greater than 0. If the two input synsets are the same, than the similarity score is 1.

5.3. Semantic Word Similarity

Since one word can be found in many synsets, each representing one sense of the word, a measure which will extend the synset similarity distance to word distance must be defined. Many Word Sense Disambiguation techniques which determine the synset (word sense) to which the word refers have been developed. However, in order to avoid the additional burden of these techniques a simplified approach was taken. Namely, the distance between two words is defined as the minimum distance between the synsets where the first word was found and the synsets where the second word was found. For example, the semantic similarity between the words *врат* (neck) and *глава* (head) is measured in the following way. As figure 5.1 illustrates, both words are found in three synsets and the similarity between each pair of synsets is calculated. The minimum distance i.e. maximum similarity between the two sets of synsets is returned as a semantic similarity between the words.



	Врат:1	Врат:2	Врат:3
Глава:1	0.428	**0.727**	0.166
Глава:2	0.142	0.533	0.142
Глава:3	0.363	0.133	0.533

Figure 5.1: Semantic similarity between the words *врат* (neck) and *глава* (head)

5.4 Semantic Text Similarity

Once the semantic word-to-word similarity measures have been defined, the next step is to define a semantic similarity measure between two text segments. The objective of this measure is to go beyond the traditional lexical matching methods and to indicate the similarity between texts at *semantic* level. Mihalcea et.al. in [37], combined the metrics of word-to-word similarity and *word specificity* into a single measure that can be used as an indicator of the semantic similarity of two texts. To determine the *specificity* of the words the authors proposed the use of the *inverse document frequency (IDF)* measure. IDF is defined as the ratio between, the total number of documents in the corpus, and the number of documents which contain the word in question. Given the word-to-word similarity and word *specificity* measures, the semantic similarity between texts T_1 and T_2 is defined as follows:

$$sim(T_1\,,T_2) = \frac{1}{2}\left(\frac{\sum_{w\,\in\{T_1\}}\big(maxSim(w,T_2) * idf(w)\big)}{\sum_{w\,\in\{T_1\}} idf(w)} + \frac{\sum_{w\,\in\{T_2\}}\big(maxSim(w,T_1) * idf(w)\big)}{\sum_{w\,\in\{T_2\}} idf(w)}\right)$$

First, for each word *w* in T_1*, the* word with highest similarity in T_2 (*maxSim(w,* T_2*)*) is identified. The same process is repeated to determine the most similar words in *T1* starting with the words in *T2*. Next, the word similarities are weighted according to their *IDF* score, summed up, and normalized with the length of each text. Finally, the resulting similarity scores are combined using average.

The similarity score has a value in the range between 0 and 1, where score of 1 indicates identical texts, and a score of 0 indicates no semantic overlap between the two input texts.

5.5. The Classification Algorithm – KNN

For the purpose of the experiment, the *K – Nearest Neighbor (KNN)* classification algorithm was considered as most suitable. The main reason for choosing this algorithm is that it allows the text similarity measure defined in the previous sections to be easily adopted. Moreover, with minimal work the algorithm can be augmented to rely on other text similarity measures, which can be used to compare the performance of the semantic text similarity measure. Finally, since the effectiveness of the algorithm is influenced only by the text similarity measure (if the other parameters are constant), it will allow the performance of the similarity measures to be evaluated and compared unambiguously.

KNN is part of the family of *nonparametric instance-based* (memory-based) methods [38]. The key idea of the algorithm is that the properties of any particular data sample ***x*** are likely to be similar to those of the data samples in the neighborhood of ***x***. Based on the distance metric specified, the algorithm finds the *K* data samples closest to the data sample given, hence the name K – Nearest Neighbor. The class of the data sample is determined by majority voting among the nearest neighbors. By default the algorithm does not require any training. However, to reduce the time needed to find the nearest neighbors, mechanisms for structuring the train samples may be introduced.

5.6 The experiment

CORPUS: For the purpose of the experiment a corpus of Macedonian news articles was used. The articles are taken from the archive of the A1 Television Website (http://www.a1.com.mk/) published between January 2005 and May 2008. As table 5.1, shows the corpus contains 9637 articles, classified in 6 categories.

PREPROCESSING: Each article was crawled from the A1 TV Website and from the HTML page only the text was extracted. All HTML tags and other noise were removed from the text. Next, each document was converted to lower case, and the punctuation and stop words were removed. The documents were represented as Python dictionaries (hash tables), where the keys represent the words contained in the document and the values represent the frequency of word in the document. This facilitates the manipulation of the documents further in the process of classification. Also, to alleviate the computation of the similarity between texts, the words in each text were sorted by their *Term Frequency – Inverse Document Frequency* (TF-IDF) value [39], and only the 20 most important words (with highest value) were retained.

SYNSET SIMILARITY METRICS: To access the Macedonian WordNet the NLTK WordNet interface module was used [29]. As discussed in the previous chapter, the Macedonian WordNet implementation developed follows the structure of the PWN database files, and therefore the same interface was also applicable to the Macedonian WordNet. In addition, the NLTK WordNet module includes implementation and allows usage of the Leacock-Chodorow (LCH) and the Wu-Palmer (WUP) synset distance metrics discussed in the previous sections.

Category	Balkan	Economy	Macedonia	Sci/Tech	World	Sport	Total
Size	1264	1053	3323	920	1845	1232	9637

Table 5.1: A1 Corpus, size and categories

CLASSIFICATION: As explained in the previous section, during the experiment the KNN classification algorithm was used. To speed up the process of classification, the training data samples were structured in *inverted index* [39]. Namely, for each word found in the corpus, a set of data samples containing the word was formed. Next, the samples were sorted by the *TF-IDF* value of the word in the text, and only the top 15 samples were retained in the index. This significantly decreases the number of comparisons needed for classification of a query text. Instead of measuring the similarity against each training sample, only the samples in the index associated with each word in the query text are considered. The value of *K*, the number of neighbors considered, was set to 5.

TEXT SIMILARITY METRICS: As a distance metric during the classification the following three text similarity metrics were used and their performance was compared:

1. Semantic text similarity metric based on the LCH synset similarity distance
2. Semantic text similarity metric based on the WUP synset similarity distance
3. Cosine similarity metric

The metrics (1) and (2) were explained in the previous sections. The Cosine similarity (3) is a classical approach for comparing text documents, where the similarity between two documents is defined as the cosine of the angle between the two document vectors. This measure is used as a base line for comparison of the performance of the other two metrics.

EVALUATION: The classification performance of each text similarity measure was evaluated in terms of the *precision, recall, and F-Measure* scores. Where, precision (P), recall (R), and F-Measure (F) are defined as follows:

$$P = \frac{number\ of\ true\ classifications\ in\ the\ class}{number\ of\ documents\ classified\ in\ the\ class},$$

$$R = \frac{number\ of\ true\ classifications\ in\ the\ class}{total\ number\ of\ documents\ in\ the\ class},$$

$$F = 2 * \frac{P * R}{P + R}.$$

The F-Measure combines the precision and recall into a single global measure of the performance.

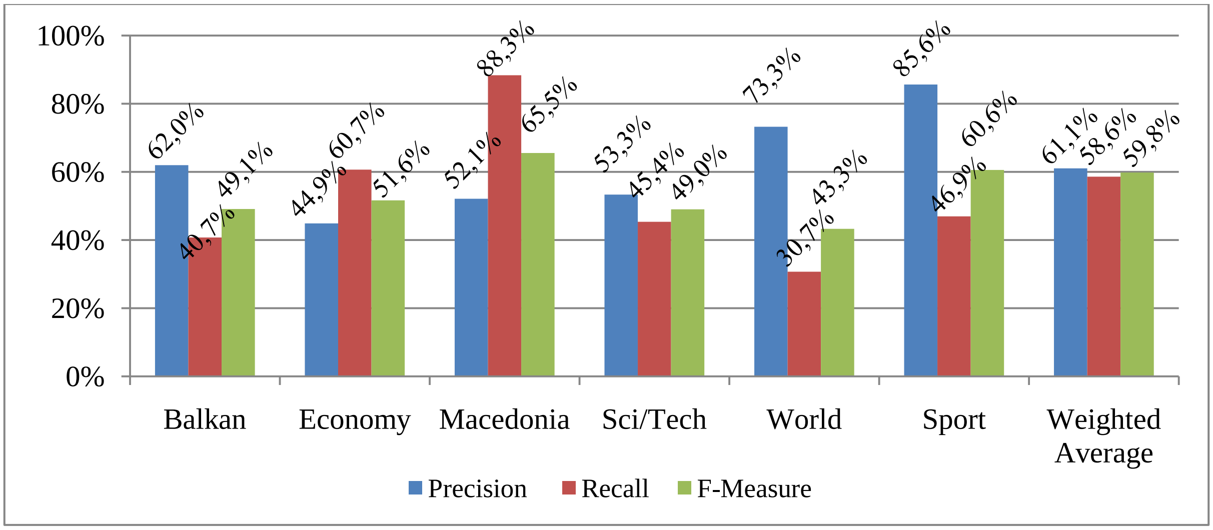


Figure 5.2: Performance of the text similarity measure based on the LCH synset similarity

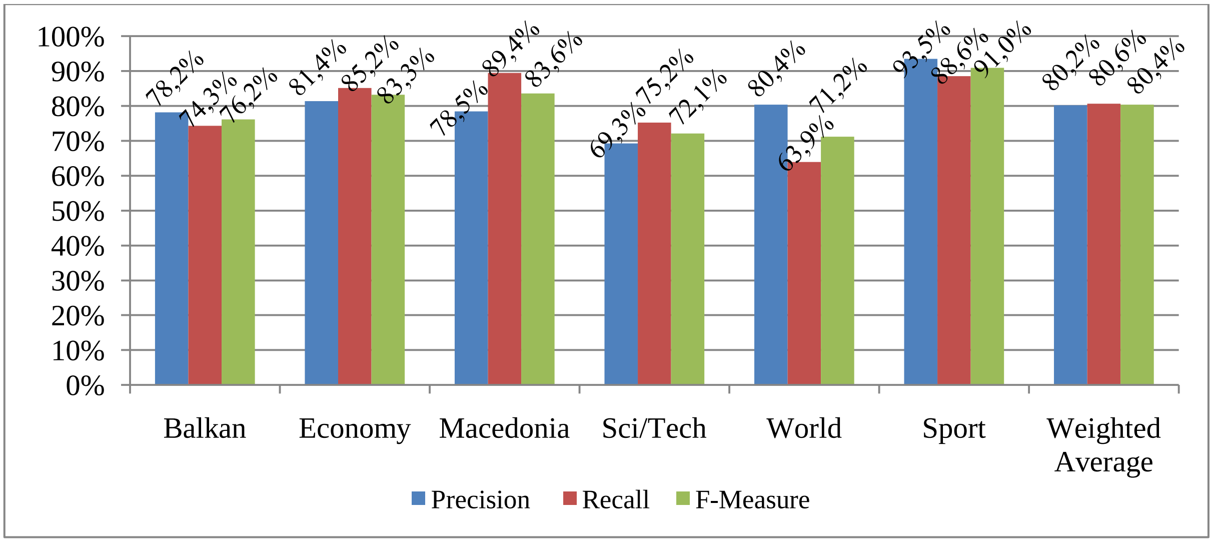


Figure 5.3: Performance of the text similarity measure based on the WUP synset similarity

Figures 5.2 and 5.3 illustrate the classification performance of the semantic text similarity measures based on the WUP and LCH synset metrics, respectively. The average of the measures is weighted by the number of samples in each category. As it can be seen from the figures, the WUP similarity significantly outperforms the LCH similarity.

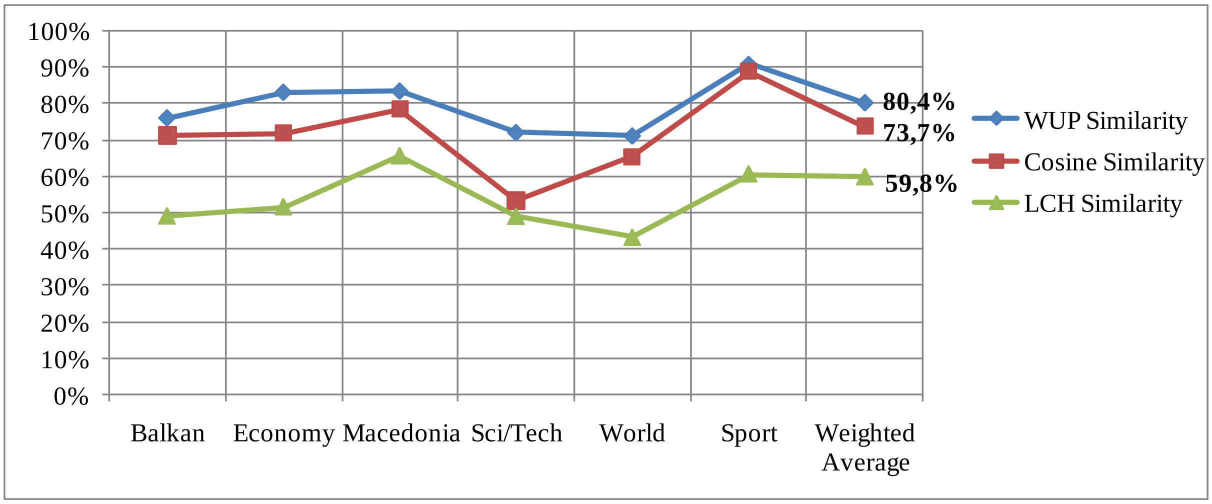


Figure 5.4: Comparison of all text similarity measures (F-measure)

Finally, figure 5.4 compares the performance of the three similarity metrics by their F-Measure score. As it can be seen from the figure, the LCH-semantic similarity fails to improve the performance of the Cosine similarity metric. The main reason for the low performance of this measure is due to its inability to calculate the similarity between words with different part of speech. The WUP-semantic similarity metric, on the other hand, has improved classification performance and outperforms both the Cosine similarity and LCH-semantic similarity metrics by ***6.7%*** and ***20.6%***, respectively. When compared to the Cosine similarity, as a baseline, this metric achieves to find more patterns in the text documents. This is especially significant in the documents from the Sci/Tech and Economy categories.

CONCLUSIONS: Although, by doing this experiment we cannot argue about the validity of the WordNet produced, we can conclude that the information encoded by the WordNet is meaningful and accurately models the real world. Moreover, we have practically shown that the Macedonian WordNet can be used to include semantics in the existing Machine Learning and Natural Language Processing algorithms and to improve their performance.

Chapter 6

Critical Evaluation

6.1 Overview

The previous chapters looked at the design and implementation of the Macedonian WordNet and the success of the project as a product. This chapter looks at the success of the overall project as an academic exercise, in terms of the aims and objectives set in the introduction of the project. In addition, there are sections which discuss what I have learnt from the project and what would I add to the project if it was repeated or extended.

6.2 Evaluation of the success of the project

In the beginning of the project, I was aware of the fact that the project is complex and requires a lot work and knowledge to be completed. The fact that I am developing something that is far beyond the standard software artifacts was a great challenge and inspiration for me. Now, after the competition of the project, I am fully satisfied with the process in which the project was developed as well as the realization of all elements of the project. Moreover, I am fully amazed by the results obtained during the implementation of both, the existing WordNet construction method and my original method for the construction. Namely, as it can be seen from the list bellow, all of the objectives envisioned in the beginning of the project were realized and satisfactory results have been produced.

- The structure of the Princeton WordNet has been successfully studied and analyzed in great detail,
- Other WordNet implementations have been explored and their approach and methodology have been studied,
- A substantial amount of research has been conducted to study and analyze the existing methods for automated WordNet construction,
- As a main objective of the project, a method for automated WordNet construction has been proposed and implemented for the purpose of the construction of the Macedonian WordNet (MWN),
- The use of WordNet for various Natural Language Processing and Machine Learning applications has been studied, and an experiment of using MWN for text classification has been conducted.

Since, all the envisioned aims have been met, it can be concluded that the project, as an academic exercise, has been successful.

6.3 What I have learnt from the project

The project has been very beneficial academic exercise. By doing the project, I have both extended my theoretical knowledge and strengthened my practical skills.

Especially, I have gained a lot of knowledge in the field of Machine Learning and Natural Language Processing. I have studied how to collect resources, manage, and use them in order to develop a WordNet. I have also studied WordNet and learnt how to use it as tool which allows semantics to be included in the existing algorithms.

Furthermore, I have learnt how to approach and solve hard problems, how to research, find the eminent authors in a field, understand the scientific resources, follow the references, and arrive to conclusions by considering the work of the others. Moreover, I have learnt that it is crucial to study and arrogate the vocabulary used in the scientific publications as well as the principles in which the problems are formulated and approached.

As a person passionate for programming, usually I focus on the implementation of the problem solutions and frequently I leave the documentation writing to be done afterwards. Thus, when writing the documentation, many times I spent a considerable amount of time to refresh my memory and reread the materials I have used during the implementation. However, in the course of this project, I have learnt that is very important to align the documentation writing programming tasks in order to be increase the productivity.

Also, I have learnt that there is no point performing a tremendous amount of valuable and important computing work, if I cannot effectively present my findings. Therefore, during the course of the project as much as on implementation I have focused on writing and expressing the concepts developed in the most suitable way.

Although, I have done a lot of projects in the other modules, this is the largest single piece of work that I have been responsible for. Therefore, I had to learn and apply new project management techniques, which I had never needed before. Moreover, I believe that part of the success is due to the way in which I have managed the time and activates during the project.

6.4. If the project were to be repeated

Looking back at the project, I would have changed some things in the way the project was developed, if I had to do it again. Some of them include the following.

First, I would put more focus on the use of parallel corpora for the development of Machine Readable Dictionary (MRD). I have spent a considerable amount of time to find, collect, and unify the existing dictionaries. I believe that it would be more beneficial to solely collect larger parallel corpora, if available, and following the same methodology produce larger MRD.

Next, I would concentrate more on the meaning of the results obtained by the application of both, the method I have proposed and the existing method for WordNet construction. I would further study and put more effort on the more effective method. Also, I would enhance my approach of combining the results, in order to more efficiently use the results obtained individually by each of the methods.

Further, if I have access to the needed resources, I would implement other existing methods for automated WordNet construction. This will allow me to compare the results obtained and further enhance the final product.

Finally, I would collect larger corpus for the purpose of the text classification experiment. Since, the classification algorithm (KNN), strongly depends on the available training samples, I believe that the use of larger corpus will significantly improve the results.

6.5 If the project were to be extended

Limited by the amount of time available to develop the project, I was not able to further study some of the ideas that came to my mind as I was working on the main aims of the project. Some of them are explained in this section.

For the purpose of development of the MRD, I have made use of parallel corpora. Namely, after the stage of word alignment I have used radical filtering to remove the incorrect alignments. However, in this way a large amount of correct word translations has been lost. Therefore, I would extend this stage by applying a more sophisticated approach. Namely, I would define the problem of determining whether one word alignment is correct as a supervised learning problem. As training examples I would use the translations found both in the word alignments and the existing dictionaries, and as features I would use the word and sentence alignment coefficients.

Next, the method that I have proposed for WN construction, based on Google Similarity Distance, uses the Google result counts in order to determine which of the candidate words are most related to the gloss, and thus express the concept in question. However, each of the candidate words is considered independently, not taking into account the semantic relatedness which exists between some of the candidate words. Namely, if I had additional time to develop the project, I would study how the candidate words can be clustered (grouped) prior to assigning them to the synset. I would consider how, based on the individual similarity between the words and the similarity of each word and the gloss, can be determined which group is most suitable to express the concept captured by the synset. In this way, I can compensate about the possible mistakes done during both, the translation of the gloss, and the measuring the semantic similarity between the candidate word and the gloss. Moreover, the probability of incorrectly assigning a group of words to a synset is much lower than the probability of incorrectly assigning an individual word.

Further on, I would like to study and implement the other applications of WordNet. During the testing stage of the project, I have defined how the similarity between two texts can be measured, and I have used this measure for text classification. In the future, I would like to investigate how this measure will influence the performance of the text clustering algorithms. Also, I would like to further study the use of WordNet for Word Sense Disambiguation. If the correct sense of the words can be determined more accurately, than the semantic similarity between them can be measured more efficiently. This will also influence the performance of the text similarity measure, and will allow the semantic similarity between texts to be measured more accurately.

Finally, I would spend some time in promoting the results achieved. I would build a Web portal which will allow users to explore and use the WordNet developed. Moreover, I would also include a guide which will explain how the Macedonian WordNet can be imported and used by other researches for the purpose of their applications. This will encourage others to involve and contribute to the progress in this area.

6.6 Conclusion

The project has been success and there is a lot of potential for future development and usage of the WordNet prototype developed. Overall it can be concluded that the techniques selected to be used during the course of the project have been adequate and succeeded to meet all of the initially envisioned objectives. Moreover, the project has been very beneficial as an academic exercise and allowed me to discover my strengths and strengthen my weaknesses. Finally, this project has helped me to identify my interests and to decide how I would like to professionally and academically develop myself in the future.

Chapter 7

References

7. References

1. Microsoft Research. (2009). *Natural Language Processing* [Online]. Available from: http://research.microsoft.com/en-us/groups/nlp/ [Accessed: 27 December 2009]

2. Lenat, D. B. (1995). CYC: a large-scale investment in knowledge infrastructure. *Communications of the ACM*. 38 (11) p. 33-38.

3. The Global WordNet Association. (2009). *Wordnets in the world* [Online]. Available from: http://www.globalwordnet.org/gwa/wordnet_table.htm [Accessed: 27 December 2009]

4. The Global WordNet Association. (2009). *Global WordNet Association Official Web Site* [Online]. Available from: http://www.globalwordnet.org/ [Accessed: 27 December 2009]

5. Fellbaum, C. Et al. (1998). *WordNet: An electronic lexical database*. Cambridge: MIT press

6. Vossen, P. (1998). *EuroWordNet: a multilingual database with lexical semantic networks*. Dordrecht: Kluwer Academic Publishers.

7. Vossen, P. Et al. (1998). *The EuroWordNet Base Concepts and Top Ontology*, Version 2, page 7

8. Gonzalo, J., Verdejo F., Chugur I. (1999). Using EuroWordNet in a Concept-based Approach to CrossLanguage Text Retrieval. *Applied Artificial Intelligence*. 13 (7) p. 647-678.

9. Tufis, D. Et al. (2001). *BALKANET: A Multilingual Semantic Network for Balkan Languages.*

10. Tufis, D., Christea, D. & Stamou S. (2004) *BALKANET: Aims, methods, results and perspectives. a general overview.* Romanian Journal on Science and Technology of Information. Special Issue on BalkaNet.

11. Pavelek, T., Pala, K. (2002). VisDic: A new Tool for WordNet Editing. *First International Global Wordnet Conference*.

12. Fišer, D. & Sagot, B. (2008). *Combining Multiple Resources to Build Reliable Wordnets.* Text, Speech and Dialogue (LNCS 2546). Berlin; Heidelberg: Springer, 2008 pp. 61-68.

13. Hiroyuki, K. & Watanabe, K.(2004). *Automatic Construction of Japanese WordNet.*

14. Changki, L. & JungYun, S. (2000). *Automatic WordNet mapping using word sense disambiguation.* In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.

15. Barbu, E. & Mititelu, B. V. (2007). *Automatic Building of Wordnets*. In Proceedings of Recent Advances in Natural Language Processing IV, John Benjamins, Amsterdam, 2007, p. 217-226.

16. Dagan, I. & Itai, A. (1994). *Word Sense Disambiguation Using a Second Language Monolingual Corpus.* In the proceedings of Computational Linguistics 1994, Vol. 20, pp. 563 – 596.

17. Cilibrasi, R. & Vitanyi, M. B. (2007). *The Google Similarity Distance.* In the proceedings of IEEE Trans. on Knowl. and Data Eng., vol. 19, pp. 370-383.

18. Keller, F. & Lapata, M. (2003). *Using the web to obtain frequencies for unseen bigrams*. In the proceedings of Computational Linguistics 2003, Vol. 29:3, pp. 459–484.

19. IDIVIDI (2010). *Idividi Macedonian Web Portal* [Online]. Available from: http://www.idividi.com.mk [Accessed: 25 March 2010]

20. 365 (2010). *365 Macedonian Web Portal* [Online]. Available from: http://www.365.com.mk [Accessed: 25 March 2010]

21. Wiki Webz (2010). *Converted Wiktionary and Wikipedia files* [Online]. Available from: http://wiki.webz.cz/dict/ [Accessed: 25 March 2010]

22. Charitakis, K. (2007). *Using parallel corpora to create a Greek-English dictionary with Uplug*. In the proceedings of Nodalida 2007, The 16th Nordic Conference of Computational Linguistics, 25-26 May 2007 in Tartu, Estonia.

23. Hao-chun, X. & Zhang, X. (2008). *Using parallel corpora and Uplug to create a Chinese-English dictionary*. Master Thesis. Sweden: KTH Royal Institute of Technology.

24. Tibor, K. & Strunk J. (2006) *Unsupervised Multilingual Sentence Boundary Detection*. In Computational Linguistics. Vol. 32, Num. 4.

25. Uplug (2010). *Uplug Official Web Site* [Online]. Available from: http://urd.let.rug.nl/~tiedeman/Uplug/ [Accessed: 25 March 2010]

26. Tiedemann, J. (2003). *Recycling Translations - Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing*. Doctoral Thesis. Studia Linguistica Upsaliensia 1

27. Tiedemann, J. (2009). *News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces* [Online]. Available from: http://urd.let.rug.nl/tiedeman/OPUS/ [Accessed: 25 March 2010]

28. Varga D., Et Al (2005). *Parallel corpora for medium density languages*. In RANLP 2005, pp. 590-596.

29. NLTK (2010). *Natural Language Toolkit - Official Web Site* [Online]. Available from: http://www.nltk.org/ [Accessed: 25 March 2010]

30. XCES (2008). *Corpus Encoding Standard for XML* [Online]. Available from: http://www.xces.org/ [Accessed: 25 March 2010]

31. Petrovski, A. (2008). *Морфолошки компјутерски речник - придонес кон македонските јазични ресурси*. Doctoral Thesis. УКиМ, ПМФ

32. Google AJAX APIs (2010). *Google AJAX APIs - Official Web Site* [Online]. Available from: http://code.google.com/apis/ajax/ [Accessed: 05 May 2010]

33. Budanitsky, A. & Hirst, G. (2001). *Semantic distance in WordNet: An experimental, application oriented evaluation of five measures*. In Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources.

34. Budanitsky, A. (1999). *Lexical Semantic Relatedness and its Application in Natural Language Processing* [Online]. Accessed from: http://www.cs.toronto.edu/

compling/Publications/Abstracts/Theses/Budanitsky-thabs.html

35. Leacock, C. & Chodorow, M. (1998). *Combining local context and WordNet similarity for word sense identification*. In Fellbaum 1998, pp. 265-283.

36. Wu, Z. & Palmer, M. (1994). *Verb semantics and lexical selection*. In Proceedings of the Annual Meeting of the Association for Computational Linguistics.

37. Mihalcea, R., Courtney, C. & Strapparava, C. (2006). *Corpus-based and Knowledge-based Measures of Text Semantic Similarity.* In Proceedings of American Association for Artificial Intelligence.

38. Stuart, R. & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. New Jersey: Pearson.

39. Manning, C. D., Raghavan, P. & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.