How to Construct Multilingual Domain Ontologies

Nitsan Chrizman Cherkassky
How to Construct Multilingual Domain Ontologies

Research Thesis

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Nitsan Chrizman Cherkassky

Submitted to the Senate of the Technion — Israel Institute of Technology
Tevet 5776 Haifa December 2015
The Research Thesis Was Done Under The Supervision of Prof. Alon Itai, in the Faculty of Computer Science.

Results from this thesis has been published in the following paper:

Acknowledgements

I would like to thank Prof. Alon Itai for all the help and support during my long research. Second, I would like to thank my parents, Ilana and Sasi Chrizman, my sister, Michal, my grandmother, Nelly Kopelevich, and the rest of my family and friends for the support along the way and for having faith all these years. Third, I would like to thank my amazing son, Jonathan, who makes it all worthwhile. Last but not least, I would like to thank my amazing husband and my source of inspiration, Michael Cherkassky. You taught me more than you will ever know.

Contents

List of Figures

Abstract 1

1 Introduction 3

2 Related Work 5
  2.1 Automatically extract ontologies from textual data or the web . . . . . . . 5
  2.2 Automatically extract ontologies from general ontologies . . . . . . . . . 6
  2.3 Multilingual ontologies . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
    2.3.1 Building multilingual ontologies . . . . . . . . . . . . . . . . . . . . 6
    2.3.2 Using multilingual ontologies . . . . . . . . . . . . . . . . . . . . . 7
  2.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

3 Data 9
  3.1 Contrasting corpora . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
  3.2 WordNet . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
  3.3 Wikipedia . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

4 Accumulating the domain’s basic concepts in the source-language 13
  4.1 Contrasting corpora . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
    4.1.1 Method . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
    4.1.2 Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
    4.1.3 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
  4.2 Wikipedia’s article titles . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
    4.2.1 The naïve approach . . . . . . . . . . . . . . . . . . . . . . . . . . 16
    4.2.2 Parent-based filtering of irrelevant concepts . . . . . . . . . . . . . 16
    4.2.3 Sub-tree intersections . . . . . . . . . . . . . . . . . . . . . . . . . 18
    4.2.4 Expanding the information and adding a learning ability . . . . . . . 20
  4.3 Results summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
5 Finding the relationships between the domain’s basic concepts in the source-language 25
   5.1 Using the “WordNet” foundation-ontology 25
   5.2 Using the “Wikipedia” database 26
      5.2.1 Naive algorithms 26
      5.2.2 BFS-based algorithm 27

6 Creating the corresponding target-language ontology 31
   6.1 Using Wikipedia’s interlinks to get the target-language Concepts 32
   6.2 Extending the target-language concepts by their target-language categories 33
   6.3 Extending the target-language concepts inside the target-language domain sub-DAG 33

7 Applying the method on a different domain 35
   7.1 Accumulating the domain’s basic concepts in the source-language 35
   7.2 Finding the relationships between the domain’s basic concepts in the source-language 36
   7.3 Creating the corresponding target-language ontology 37
   7.4 Conclusions 37

8 Summary, Conclusion and Future work 39
   8.1 Summary 39
   8.2 Conclusions 39
   8.3 Future work 40

Hebrew Abstract i
List of Figures

4.1 Parents-based method, version 1: Recall vs. Precision . . . . . . . . . 17
4.2 Parents-based method, version 2: Recall vs. Precision . . . . . . . . . 19

5.1 Example for a sub-DAG of D. The concepts in the quadrilaterals are the categories, and the concepts in the circles are the articles. the connection represented by an arrow is "part-of" (as described in section 3.3) . . . . . 28
Abstract

In information science, an ontology is a structural framework for organizing information, which represents knowledge as a set of concepts and the relationships between them. During the last decades, ontologies in general, and multilingual ontologies in particular, have become the most popular and widespread means of knowledge representation, and have been used in many fields as a tool for storing and representing knowledge.

Our research focuses on automatic construction of multi-lingual domain-ontologies, i.e., creating a DAG (directed acyclic graph) consisting of concepts relating to a specific domain and the relations between them. The concepts of an ontology are expressed as words and phrases of a language. A multi-lingual ontology consists of several single language ontologies and the connection between corresponding concepts.

Our final result is a method which, given a domain, produces a multilingual ontology that represents the main concepts within the domain, along with their attributes and the relationships between those concepts. The methodology we developed includes developing, using a given domain and relying on a foundation ontology, a domain-ontology in a source language (the main language) and, based on that, develop a corresponding domain-ontology in a target language (the secondary language).

We examined three data sources: Contrasting corpora, WordNet and Wikipedia, and compared the use of each source. We then considered three tasks:

1. Accumulating the domain’s basic concepts—Gathering a large set of “relevant concepts”, concepts which are closely related to the domain core.

2. Developing the domain-ontology in the source language—Finding the connections between the concepts, that were obtained in the first task.

3. Developing the corresponding domain-ontology in the target language—Developing an ontology in the target language based on the ontology in the source language we had created in the previous tasks.

For each task we tried to rely on one or more databases from those presented above, and chose the one which gave the best results.

For the first task we tried to use the contrasting corpora and Wikipedia. While the contrasting corpora provided unsatisfactory results, the Wikipedia results were satisfying, especially when (1) expanding the information we used, from only the structure of
Wikipedia, to the words appear in the Wikipedia’s abstracts, and (2) adding a learning ability to identify related concepts by the words in their abstracts.

For the second task we tried to use WordNet and Wikipedia. It turned out that WordNet lacked many important concepts and connections. However, the Wikipedia database turned out to be suitable.

For the third task we used only Wikipedia, whose results were satisfactory. In all three tasks, Wikipedia proved to be the best choice. Thus, during the process, our methods rely only on it.

Our domain example on which we perform the research is organized crime, and we chose, for this particular work, to use the English and Hebrew languages as the source and target languages.
Chapter 1

Introduction

In information science, an ontology is a structural framework for organizing information, which represents knowledge as a set of concepts and the relationships between them. This model can be represented as a graph, in which the nodes are the concepts, and the edges are the relations between them. During the last decades, ontologies have become the most popular and widespread means of knowledge representation, and have been used in information retrieval [30], NLP applications [7], software engineering for software modelling [2], in biomedical informatics for genes representation [1], in business for online customer complaint management [12] and more.

A foundation-ontology [23] is a model which describes common concepts, and is as general as possible, for example Word-Net. In contrast, a domain-ontology [18] models a specific domain, or part of the world, by representing the particular meanings of terms and relations among them, as they apply to that domain. As a result, in a restricted domain the word-sense ambiguity is reduced. For example, there are many concepts which refer to “Mercury” the element, the planet, the Roman God and more, but in the specific domain of Chemistry only the element concept is relevant. Our research focused on domain-ontologies.

In recent years, due to globalization, multilingual exchangeability has gained importance. Consequently, multilingual ontologies were developed and have become an important tool for mapping concepts and relations between languages. There are many reasons to develop a multilingual ontology. We now describe two which motivated us to deal with this model:

- Providing cross-lingual expressions: Many common and useful expressions cannot be literally translated, and a semantic translation is required. Thus, a multilingual ontology can be a reliable and useful database for this purpose.

- Providing ontologies in resource restricted languages: Automatic engineering of ontology is naturally based on large corpora, in order to identify the relevant concepts and their relations. The larger the resources, the more accurate and wide-scale is the ontology. Thus, in resource restricted languages it is hard to
automatically construct ontologies. Therefore, developing a multilingual ontology may provide a workaround solution to this problem.

In our research we have developed a method which, given a domain, produces a multilingual ontology that reveals the main concepts within the domain, along with their attributes and the relationships between those concepts. The methodology we developed includes developing, using a given domain and relying on a foundation ontology, a domain-ontology in a source language (the main language) and, based on that, develop a corresponding domain-ontology in a target languages (the secondary language). We concentrate in our research only on “IS-A” relations, i.e., membership and subset. Other ontologies discuss additional relations such as antonyms and meronyms.

The languages we used for our research are English (source language) and Hebrew (target language). Our domain example on which we perform the research is Organized Crime. We include in this domain all aspects of crime which are organized, for any motivation: financial, ideological (terror), etc.

For this purpose we divided the challenge into the following three tasks:

- Accumulating the domain’s basic concepts;
- Developing the English domain-ontology;
- Developing the corresponding Hebrew domain-ontology.

In order to justify the generality of this method, we examined the entire method on another domain – ”Vehicles” – as well.

The main contribution of this work is the framework it provides for constructing a multilingual-domain-ontology. This framework contains the entire process, from gathering the main concepts in the source language, until the desired output of a full multilingual-domain-ontology contains both source and target languages. In this work, different approaches to handle this task are presented, along with their results and analyses.

The rest of this work is organized as follows. Section 2 gives background information on previous related efforts. Section 3 reviews the databases that were examined in this work. Sections 4 and 5 presents several methods we developed in order to accumulate the domain’s basic concepts and to find the relationships between them. Section 6 deals with the creation of the corresponding Hebrew ontology. Section 7 describes the application of the method on different domain. Section 8 contains a summary, conclusions and future work.
Chapter 2

Related Work

Many researches worked on building single language and multilingual ontologies. We now review the most significant ones, divided into three main subjects.

2.1 Automatically extract ontologies from textual data or the web

Biebow and Szulman [3] presented TERMINAE, a computer-aided knowledge engineering tool which helps building an ontology based on relevant corpus. Its originality is to integrate linguistic and knowledge engineering tools. The linguistic engineering part allows the definition of terminological forms from the study of term occurrences in a corpus. The knowledge engineering part involves knowledge-base management for the ontology.

Velardi, Missikoff and Basili [31] described a text mining technique to aid an Ontology Engineer to identify the important concepts in a Domain Ontology, based on relevant concepts. They used natural language processing tools for two tasks: discovering of terms that are good candidate names for the concepts in the Ontology, and identification of taxonomic relations among these terms.

Navigli and Velardi [18] presented a method and a tool, OntoLearn, aimed at the extraction of domain ontologies from Web sites. OntoLearn first extracts a domain terminology from available documents. Then, complex domain terms are semantically interpreted and arranged in a hierarchical fashion. Finally, a general-purpose ontology, WordNet, is trimmed and enriched with the detected domain concepts.

Maedche and Staab [16] developed a general architecture for discovering conceptual structures and engineering ontologies. They used a generalized association rule algorithm that does not only detect relations between concepts, but also determines the appropriate level of abstraction at which to define relations.
2.2 Automatically extract ontologies from general ontologies

Wu and Weld [32] developed the Kylin Ontology Generator (KOG), an autonomous system that builds a rich ontology by combining Wikipedia infoboxes with WordNet using statistical-relational learning. It predicts subsumption relationships between infobox classes while simultaneously mapping the classes to WordNet nodes. KOG also maps attributes between related classes, allowing property inheritance.

Syed, Finin and Joshi [27] described the use of Wikipedia and spreading activation to find generalized or common concepts related to a set of documents using the Wikipedia article text and hyperlinks. They started their experiments with the prediction of concepts related to individual documents, and extended them to predict concepts common to a set of related documents.

Suchanek, Kasneci and Weikum [26] presented YAGO, a large ontology with high coverage and precision. YAGO was automatically derived from Wikipedia and WordNet. It comprises entities and relations, and contains more than 1.7 million entities and 15 million relations. These include the taxonomic Is-A hierarchy as well as semantic relations between entities. The facts for YAGO have been extracted from the category system and the infoboxes of Wikipedia and have been combined with taxonomic relations from WordNet. A powerful query model facilitates access to YAGO’s data.

Hepp, Bachlechner and Siorpaesshow [17] show that standard Wiki technology can be easily used as an ontology development environment for named classes, prove that the URIs of Wikipedia entries are surprisingly reliable identifiers for ontology concepts, and demonstrate the applicability of this approach in a use case.

Melo and Weikum [8] investigated how entities from all editions of WordNet as well as Wikipedia can be integrated into a single coherent taxonomic class hierarchy. They relied on linking heuristics to discover potential taxonomic relationships, graph partitioning to form consistent equivalence classes of entities, and a Markov chain-based ranking approach to construct the final taxonomy. This results in MENTA (Multilingual Entity Taxonomy).

2.3 Multilingual ontologies

2.3.1 Building multilingual ontologies

Trojahn, Quaresma and Vieira [28] propose a framework for mapping multi-lingual Description Logics (DL) ontologies. First, the DL source ontology is translated to the target ontology language, using a lexical database or a dictionary, generating a DL translated ontology. The target and the translated ontologies are then used as input for the mapping process. A DL mapping ontology is generated as result of this process.

Trojahn, Quaresma, Vieira [29] describe an API for multi-lingual matching that
implements two strategies, direct translation-based and indirect. The first strategy
considers direct matching between two ontologies (i.e., without intermediary ontologies),
with the help of external resources, i.e., translations. The indirect alignment strategy
is based on composition of alignments. They evaluated these strategies using simple
string similarity based matchers and three ontologies written in English, French, and
Portuguese.

Chaves and Trojahn [5] present Hontology, a multilingual ontology for the hotel
domain. Hontology has been manually created and its current version supports the
English, French and Portuguese languages. Each concept and property of Hontology are
manually annotated with different labels in these three languages. Although for dealing
with the huge source of knowledge at the web scale, automatic methods for creating
and populating ontologies are required, Hontology can be seen as a starting point to
these approaches.

Garcia, Ponsoda, Cimiano, Gomez, Buitelaar and McCrae [10] present a vision of
a multilingual Web of Data. They discuss challenges that need to be addressed to
make this vision come true, and the role that techniques, such as ontology localization,
ontology mapping, and cross-lingual ontology-based information access and presentation,
will play in achieving this. Further, they propose an initial architecture and describe a
roadmap that can provide a basis for the implementation of this vision.

Lauser, Wildemann, Poulos, Fisseha, Keizer and Katz [14] introduces a comprehen-
sive framework for building a domain-specific ontology, based on a focused web crawler
the well-established thesaurus.

2.3.2 Using multilingual ontologies

Declerck, Perez, Vela, Gantner and Manzano-Macho [9] implemented a platform that
allows the user to upload a specific ontology, to select labels of the ontology and the
language to which this label should be translated. Once the user has made his selections,
the systems accesses the EuroWordNet and Wikipedia databases for finding if the
selected term is encoded in the resources and displays the results of the search to the
user, who can then decide if the suggestions made by EWN or Wikipedia are appropriate.

Lin and Krizhanovsky [15] developed automatic translator for obtaining correct
matching pairs in multilingual ontology matching. In the case study, the problem
entity is a task of multilingual ontology matching based on Wiktionary data accessible.
Ontology matching results obtained using Wiktionary were compared with results based
on Google Translate API.

Guyot, Radhouani and Falquet [11] explored a translation-free technique for multi-
lingual information retrieval, based on an ontological representation of documents and
queries. They use multilingual ontologies to map terms, documents and queries to their
corresponding concepts. This way there is no dependency on automatic translators.
2.4 Summary

The first group of works (Section 2.1) concentrated on extracting ontologies, mostly from different databases as textual data or the web.

The second group of works (Section 2.2) took advantage of general existing ontologies (mostly Wikipedia and WordNet), and tried to enrich them, in order to use the derived information for other purposes than building a domain ontology.

The third group of works (Section 2.3) concentrate on multilingual ontologies. They are based on different kinds of tools, most of them dictionaries, in order to produce the multilingual ontologies.

In our work we tried to develop a new approach, which uses a foundation ontology in order to extract a domain ontology in the source language, and based on it, in the target language as well. In contrast, the works described above, concentrated on just one of these two tasks, and handled it with different approaches. As can be seen from the works described above, the idea of construct a multilingual ontology from Wikipedia, and the idea of concentrating on building a multilingual domain-ontology, have not been researched enough separately, and not at all combined together. Thus, in this work we developed innovative ways to deal with these new aspects, from a different direction and as a one combined method.
Chapter 3

Data

In our work we examined three data-bases we used for developing the methods describes in the next chapter. The examined data-bases were:

- Contrasting corpora
- WordNet
- Wikipedia

The results obtained from the first two databases were unsatisfactory, while Wikipedia proved beneficial.

3.1 Contrasting corpora

This database contained two corpora: an in-domain and an out-of-domain. They were collected automatically by a crawler, in order to acquire an appropriate database, of the domain’s relevant concepts. The first corpus (called $C_{dom}$) was dedicated to the subject domain, and it contained 800 articles which discuss various topics related to organized crime. These articles were automatically collected from web-sites dealing with organized-crime, relevant sections in online newspapers, works and articles in this domain, etc. The second corpus (called $C_{gen}$) was a general one. It contained 8304 articles on a wide range of topics and was used as a control group. Its articles were collected from sources similar to the first corpus, except that they were not constrained to the subject domain.

3.2 WordNet

WordNet is a freely and publicly available semantic dictionary of English, developed at Princeton University. It groups English words into sets of synonyms called synsets, and records the various semantic relations between these synonym sets. Thus, this database is often used, as an ontology to support automatic text analysis, artificial intelligence
applications, and NLP researchers [26, 24]. I used the JWI (the MIT Java WordNet Interface) which supports access to WordNet versions 1.6 through 3.0, among other related WordNet extensions.

3.3 Wikipedia

Wikipedia is a multilingual, web-based, free-content encyclopedia project. Its vast scope (4,252,811 English articles in June 2013), its diversity, and its data organization structure make it a very attractive database to NLP researchers. In order to use the Wikipedia database, we relied on the DBpedia knowledgebase project [4], which contains the information of the English Wikipedia (June 2012 version) in pre-processed files. Furthermore, it contains interlinks for most of the articles, from its English version to the corresponding article in up to 111 languages (including Hebrew).

We may view the Wikipedia database as a directed acyclic graph (DAG), $W$, which has two kinds of nodes:

- Internal nodes (Wikipedia’s categories) – These nodes have one or more parents (categories) and one or more descendants (categories / articles). In this DAG there is only one root category “Category: Contents”) with no parent.

- Leaves (Wikipedia’s Articles) – The heart of Wikipedia. Each leaf contains information about a specific subject. In the first methods in Chapter 4 (Accumulating the domain’s basic concepts), only the articles’ titles were used as labels (for example: “Sicilian Mafia”), but it was expanded to use more information in the last method. Articles can have one or more parents (categories) but do not have any children. Each of the article parents is declared as a “category of the article”.

The Wikipedia’s DAG described above can be treated as an ontology in the natural way – the nodes of the DAG are the nodes of the ontology, and the edges of the DAG are the connections between the items of the ontology. This ontology contains only two kinds of connections:

- Object/concept which is part of a group (article-category);

- Group which is sub-group of a larger group (category-category).

Most of the methods described in Sections 4.2 and 5.2 deal with the sub-DAG, $D$, rooted at the node that represents the domain. in our example: “Category: Organized crime”; i.e., the node corresponding to “Organized crime” and all nodes reachable from it. To examine the performance of these methods, we collected a set of articles from Wikipedia, which contains manually classified articles from the “organized crime” sub-DAG ($D$) as follows: We created a balanced database which contains 290 articles (from $D$), out of which half (145) were relevant to the subject, and the rest were not. Using balanced data
facilitates both learning and evaluation. If the data is not balanced, the simple heuristic of always choosing the majority section may yield good results on the test-data, but will not apply to other data. The initial accumulated database contained 319 articles (initially it was 320, but one of the concepts was a category), which were collected randomly from the set of the concepts in $D$. These articles were manually classified as relevant(145)/irrelevant(174). Giving the initial database (319 articles), a database of 290 articles is the largest balanced database we could created.

An article was classified as relevant if its title met at least one of the following criteria:

- Name of a person who is famous mainly because his relations to the domain world. Famous people who are indirectly related to this domain, weren’t classified as relevant. For example, In the examined domain, “organized-crime”, presidents and singers, weren’t classified as relevant.

- Names of organizations that operate mainly in the domain.

- Expressions (one word or collocation), whose meaning is related to the domain scope. For example, In the examined domain, ”organized-crime”, “Money Laundering”, “mafia”, etc.

However, articles whose titles are branches of the domain, but are mainly related to other domain, were classified as irrelevant, even if they satisfied the above criteria. For example, In the examined domain, ”organized-crime”, art dealing with organized crime, such as crime-literature and crime-movies, were classified as irrelevant.

Table 3.1 summarizes which databases we tried to rely on for each task.

In the following sections we will describe the method we used in order to achieve the goals, along with their evaluations and results.
Chapter 4

Accumulating the domain’s basic concepts in the source-language

In this task, the main target was to gather a large set of “relevant concepts”, which are closely related to the domain core. These objects are language-independent and were accumulated in the source language. The definition of “closely related” is problematic. We do not have a defined and formal measure to estimate the quality of the concepts, and worse, this “quality” is often a subjective perspective. Thus, one option to handle this problem is by giving each concept a grade for its relevancy. The second option, which we chose because it was more easy to implement and to use, is to give a binary grade: relevant/irrelevant. The concepts which fall in the middle, and are not significantly relevant, caused problems in the methods in discussion.

In this task we employed statistical methods. For each method, the performance was estimated based on its false-positive and false-negative results. The concepts accumulated in this task were used in the second task (Chapter 5), as the foundation of the ontology.

4.1 Contrasting corpora

4.1.1 Method

In this section we created a known supervised method, which, based on pre-classified articles, selected the expressions most relevant to the domain. The classified articles were taken from the contrasting pair of corpora, described in Section 3.1, and were used to create a set of collocations. Hopefully most of them will be multi word expressions – expressions, whose meaning is not compositional, i.e., they are not predictable from the meanings of the individual lexemes. We concentrated only on collocations, since these expressions are rarely ambiguous, and therefore are easy to classify. If this method would have yielded sufficient results, we would have extended it to support single words as well. These expressions were meant to be used as “relevant concepts” for building
the ontology in the next task. The method in discussion is based on two well known feature-selection techniques, which were tested on the database:

**PMI (Pointwise mutual information) [6]**
The PMI of a pair of outcomes $x$ and $y$, measures the discrepancy between the probability of their co-occurrence given their joint distribution and their individual distributions, assuming independence.

$$PMI(x, y) = \log \frac{p(x, y)}{p(x)p(y)}$$

**TF-IDF (term frequency – inverse document frequency) [22, 13, 21]**
Input: term $t$
Output: a value which represents the relevance of term $t$ to the domain subject

- $TF(t, D) = \text{frequency of } t \in D$
- $IDF(t, D) = \log \left( \frac{|D|}{\sum_{d \in D} \delta(t \in d)} \right)$
- $\text{TF-IDF}(t, D) = TF \times IDF$

As mentioned above, the desired results (collocations) should satisfy two criteria:
(1) be a collocation (assured by the PMI), and (2) be relevant to the domain (assured by TF-IDF). In order to accumulate such expressions, a combination between the two kinds of techniques was performed. We combined between these two measurements described above in the following way:
Choose $n$ collocations with the highest PMI value, and then choose $k$ collocations out of them with maximum TF-IDF value.

### 4.1.2 Evaluation

We conducted a series of experiments in order to evaluate the combination of the described indicators. During the experiments, different values of $k$ and $n$ were examined ($k \in \{20, 40, 60, 80, 100\}$, $n \in \{4000, 6000, 8000, 10000\}$). These values were sufficiently large in order to provide a statistically stable results, and small enough to allow a manual classification of the results. Furthermore, as can be seen from the results section, there were no meaningful differences between the experiments’ results, a fact that supported our decision to rely only on these values of $k$ and $n$.

For each experiment we calculated only the "true-positive" value. The reason we did not calculate the "false-negative" values, is that we did not have enough information to perform this calculation. In order to correctly calculate the "false-negative" values, we had to be able to calculate how many relevant expressions were not found during the experiments. Since we did not have an entire set of the relevant concepts, these values could not be calculated.
Table 4.1: The results of the “online articles” method with 5% confidence interval

<table>
<thead>
<tr>
<th></th>
<th>n</th>
<th>4000</th>
<th>6000</th>
<th>8000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.35 (±0.21)</td>
<td>0.4 (±0.21)</td>
<td>0.4 (±0.21)</td>
<td>0.4 (±0.21)</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>0.45 (±0.15)</td>
<td>0.42 (±0.15)</td>
<td>0.43 (±0.15)</td>
<td>0.42 (±0.15)</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>0.5 (±0.13)</td>
<td>0.517 (±0.13)</td>
<td>0.516 (±0.13)</td>
<td>0.517 (±0.13)</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>0.44 (±0.11)</td>
<td>0.45 (±0.11)</td>
<td>0.46 (±0.11)</td>
<td>0.48 (±0.11)</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0.43 (±0.1)</td>
<td>0.42 (±0.1)</td>
<td>0.42 (±0.1)</td>
<td>0.43 (±0.1)</td>
<td></td>
</tr>
</tbody>
</table>

For $k \in \{20, 40, 60, 80, 100\}$ and $n \in \{4000, 6000, 8000, 10000\}$:

- the top $n$ expressions, according to the PMI values were retrieved.
- the top $k$ expressions out of the $n$ expressions, according to the TF-IDF values were retrieved.
- The $k$ expressions were classified manually (by the author) as relevant/irrelevant.
- The precision (number of true positive) was computed along with the confidence interval.

4.1.3 Results

As can be seen from Table 4.1, we did not reach a satisfactory result. The highest true-positive value ($0.517 \pm 0.13$) is still pretty low and is not statistical significant.

We assume that the poor results are due to the diversified database. We classified the articles roughly, but there were many articles whose relevance was uncertain. For example, in the examined domain, articles containing gossip about a mobster’s wife, or a story about a police officer dealing with organized crime. When such articles belong to the domain database, the results are affected. Consequently, we decided to abandon this method.

4.2 Wikipedia’s article titles

In this experiment we treated each Wikipedia article as a concept, and tried to choose the ones that belong to the domain. Based on the structure described above, our goal was to find the domain’s relevant articles. The titles of the resulted articles will be used as the ontology concepts. In the described methods, we concentrated only on $D$ (the domain sub-DAG), and ignored the articles in $W - D$. The large amount of articles in $W - D$ prevented us from classifying each article separately, but based on the results and on the Wikipedia structure, it is reasonable to assume that most of the desired and relevant articles belong to $D$.  

15
We tried four methods as described below. Each relied on the conclusions of the previous ones.

4.2.1 The naïve approach

The motivation: The naïve assumption is that every leaf (article) of $D$ should be a relevant concept, because it is an offspring of the category that represents the domain.

4.2.1.1 Evaluation

To evaluate this assumption we randomly chose 319 articles from $D$ and classified them as described in Section 3.3.

4.2.1.2 Results and conclusions

Out of the 319 articles, only 174 (55%) were relevant. A closer examination revealed that, unfortunately, $D$ contains many articles that deal with topics such as crime-films, thriller-authors, presidents, and they are irrelevant. Thus, this approach did not provide satisfactory results, but led us to a new challenge: the new goal was to correctly and efficiently separate the relevant leaves of $D$ from the irrelevant ones. From this point on, we looked only at these articles and tried to classify them correctly, using the tools Wikipedia provides.

4.2.2 Parent-based filtering of irrelevant concepts

In this step, we tried to classify the articles of $D$, based on their parents (categories), i.e., nodes that have a link to the article.

4.2.2.1 Parents-based method, version 1

Find-relevant-concepts (factor $\mu$):

Initialization

Let $Parents(a)$ be the set of parents of article $a$

Article $a$ is relevant if and only if

$$\mu \times |Parents(a) \cap D| > (1 - \mu) \times |Parents(a) - D|$$

Evaluation

We performed the method “find-relevant-concepts” for each value of $\mu$ in $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$. The parameter $\mu$ was used in order to control the weight of the relevant/irrelevant parents (when $\mu = 0.5$ we get an equal weight). It was an essential parameter, because it allowed us to control the importance of each group. For
Experiment number | Factor | Recall | Precision | F-score |
--- | --- | --- | --- | --- |
1 | 0.1 | 0.14 | 0.74 | 0.24 |
2 | 0.2 | 0.15 | 0.75 | 0.25 |
3 | 0.3 | 0.2 | 0.77 | 0.32 |
4 | 0.4 | 0.27 | 0.8 | 0.4 |
5 | 0.5 | 0.29 | 0.8 | 0.43 |
6 | 0.6 | 0.43 | 0.81 | 0.56 |
7 | 0.7 | 0.58 | 0.78 | 0.67 |
8 | 0.8 | 0.64 | 0.72 | 0.68 |
9 | 0.9 | 0.74 | 0.60 | 0.66 |
10 | 0.95 | 0.76 | 0.55 | 0.64 |

Table 4.2: The results of “Parents-based method, version 1”. For each factor $\mu$, the tables contains the values for recall, precision and F-score.

Figure 4.1: Parents-based method, version 1” : Recall vs. Precision

example, we were able to examine the option that one relevant parent is enough in order to classify the article as relevant, and on the contrary, the option that one irrelevant parent is enough in order to classify the article as irrelevant. The set of articles which have been used in the function “find-relevant-concepts” were the 290 manually classified articles described in Section 3.3. Finally, we defined the following parameters:

- Recall: \( \frac{\text{True-positive}}{\text{True-positive} + \text{false-negative}} \)

- Precision: \( \frac{\text{True-positive}}{\text{True-positive} + \text{false-positive}} \)

- F-score: \( \frac{2}{\text{Recall} + \text{precision}} \)

And calculated recall, precision and F-score for each experiment.

**Results**

Table 4.2 and Figure 4.1 summarize the results for this method.
Conclusions
The results were not yet satisfactory. The value of $\mu=0.8$ yielded the best result, however its F-score=0.68, is lower than expected. We recorded a large percentage of false-negatives, due to categories like year of birth, year of death, people by nationality, etc. These categories appear as parents of relevant articles, and affected the results.

4.2.2.2 Parent-based method, version 2

Find-relevant-concepts-expanded (factor $\mu$):

Find relevant categories:
Let $R = \text{Find-relevant-concepts} (\mu)$, and let $\overline{R} = D - R$.
For each category $c \in D$:
Let $G = \text{the descendants of } c$:
$c$ is relevant if and only if $|G \cap R| > |G \cap \overline{R}|$.

Find relevant leaves:
For each article $a$ in $D$:
$a$ is relevant if at least one of its parents was marked as relevant in “Find relevant categories”.

Evaluation Similarly to the evaluation of section 4.2.2.1, we evaluated the method for each factor $\mu$ in $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$, by performing 10 independent experiments for the 10 different values.

Results
Table 4.3 and Figure 4.2 summarize the results for this method.

Conclusions The results were still unsatisfactory. The best result received with $\mu=0.8$, and its F-score was 0.65, which is lower than our goal, and lower than the basic method. Thus, the expanded method did not improve the results, and even performed worse.

4.2.3 Sub-tree intersections
In this step we tried to find other sub-DAGs in the Wikipedia tree to help us correctly classify $D$ concepts, using their intersection with $D$. For example, we expected that
<table>
<thead>
<tr>
<th>Experiment number</th>
<th>Factor</th>
<th>Recall</th>
<th>Precision</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>0.15</td>
<td>0.93</td>
<td>0.26</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>0.18</td>
<td>0.9</td>
<td>0.3</td>
</tr>
<tr>
<td>3</td>
<td>0.3</td>
<td>0.23</td>
<td>0.88</td>
<td>0.37</td>
</tr>
<tr>
<td>4</td>
<td>0.4</td>
<td>0.44</td>
<td>0.88</td>
<td>0.59</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>0.45</td>
<td>0.88</td>
<td>0.6</td>
</tr>
<tr>
<td>6</td>
<td>0.6</td>
<td>0.49</td>
<td>0.83</td>
<td>0.62</td>
</tr>
<tr>
<td>7</td>
<td>0.7</td>
<td>0.58</td>
<td>0.69</td>
<td>0.63</td>
</tr>
<tr>
<td>8</td>
<td>0.8</td>
<td>0.63</td>
<td>0.68</td>
<td>0.66</td>
</tr>
<tr>
<td>9</td>
<td>0.9</td>
<td>0.76</td>
<td>0.55</td>
<td>0.64</td>
</tr>
<tr>
<td>10</td>
<td>0.95</td>
<td>0.76</td>
<td>0.55</td>
<td>0.64</td>
</tr>
</tbody>
</table>

Table 4.3: The results of "Parents-based method, version 2". For each factor \( \mu \), the tables contains the values for recall, precision and F-score.

Figure 4.2: Parents-based method, version 2" : Recall vs. Precision
the intersection of $D$ (in our example: “Category: Organized-crime” sub-tree) and “Category: Movies” sub-tree, contain mainly movies about organized crime. Thus, we hoped that removing the articles in the intersection from the relevant concepts set, would bring us closer to our goal. The results did not meet our expectations, and they were even worse. For example, many crime movies did not belong to the intersection, and this intersection also contained many concepts that were not movies. We suspect that the fact that Wikipedia has a large number of authors affected the consistency of the database’s structure.

4.2.4 Expanding the information and adding a learning ability

Due to the unsatisfactory results of the previous methods, we decided to expand the information we relied on in the classification problem, and to use a learning mechanism in order to correctly classify the Wikipedia’s articles. This method presents two additional ideas. First, we inserted significant extra information to the method: the abstracts of the articles. Each article in Wikipedia contains an abstract, a short summary which gives the highlights of the topic, and therefore probably contains the relevant expressions with high frequency. The second additional concept was the learning process. In contrast to the previous methods, which were unsupervised, in this method, a set of articles were manually tagged as relevant/irrelevant, and were used as a learning and testing sets.

For this task, We tried to use two different supervised algorithms, and compared between their results. The first algorithm is the well known “Support vector machine” (SVM), and the second algorithm was developed especially for this research.

4.2.4.1 Using SVM as supervised learning model

Support Vector Machines (SVM) are machine learning approaches for solving two-class pattern recognition problems. SVMs are well known for their good generalization performance, and have been applied to many pattern recognition problems.

In this task we tried to use SVM as a supervised learning model in order to correctly classify which concepts are relevant to the domain in discussion, and which are not. For this purpose, the abstracts of the articles were treated as bags of words. For each abstract, the number of occurrences of each word were gathered, and together with the abstract classification (relevant/irrelevant) were used as an input for the SVM model.

Evaluation

- We performed two experiments:

  1. In the first experiment, we used, as input for the SVM, the number of occurrences of each word in each abstract. Additionally, as part of this experiment, we performed the method for each value of the threshold $\theta$
Table 4.4: The results of “using SVM as supervised learning model” - first experiment. For each threshold-parameter ($\theta$) the table contains the values for average accuracy with 5% confidence interval and the standard deviation.

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>Avg. Accuracy</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.796 ($\pm$ 0.107)</td>
<td>0.042</td>
</tr>
<tr>
<td>1</td>
<td>0.8 ($\pm$ 0.107)</td>
<td>0.059</td>
</tr>
<tr>
<td>2</td>
<td>0.8 ($\pm$ 0.106)</td>
<td>0.059</td>
</tr>
<tr>
<td>3</td>
<td>0.804 ($\pm$ 0.105)</td>
<td>0.064</td>
</tr>
<tr>
<td>4</td>
<td>0.807 ($\pm$ 0.105)</td>
<td>0.063</td>
</tr>
<tr>
<td>5</td>
<td>0.8 ($\pm$ 0.107)</td>
<td>0.068</td>
</tr>
<tr>
<td>6</td>
<td>0.789 ($\pm$ 0.109)</td>
<td>0.071</td>
</tr>
<tr>
<td>7</td>
<td>0.789 ($\pm$ 0.109)</td>
<td>0.058</td>
</tr>
</tbody>
</table>

Table 4.5: The results of “using SVM as supervised learning model” - second experiment. The table contains the values for average accuracy with 5% confidence interval and the standard deviation for $\theta = 0$ with $TF-IDF$ values.

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>Avg. Accuracy</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.8 ($\pm$ 0.107)</td>
<td>0.038</td>
</tr>
</tbody>
</table>

between [0,7]. For each value $\theta$, we took into consideration only the words which occurred more than $\theta$ occurrences in the abstract.

2. In the second experiment, we used, as input for the SVM, the $TF-IDF(t,d)$ value of each word $t$ in each abstract $d$. Since the $IDF(t)$ value of word $t$ is a constant, the measurement we used is: $\frac{TF(t,d)}{|d|}$. We performed this experiment for $\theta = 0$

- We tested our method on 270 articles, out of which 135 were relevant to the subject and the rest, 135 articles, were irrelevant
- In order to establish the statistical significance of the results, a 5-fold-cross-validation method was performed.

Results

Table 4.4 summarizes the results for the first experiment of this method, and Table 4.5 summarizes the results for the second experiment of this method. As can be seen, this method yielded a relatively high accuracy for all of the values of $\theta$ in both experiments. In the second experiment the results were slightly better, but there is no specific value which was statistically better than the others.

4.2.4.2 Using a self developed supervised learning model

In this algorithm, the abstracts of the articles were also treated as bags of words. In the first step, the learning process, we tried to calculate which words appeared significantly
more often in the relevant articles’ abstracts (“relevant-words”), and which words appeared significantly more in the irrelevant articles’ abstracts (“irrelevant words”). During the second step, the testing process, an article was classified as relevant if its abstract, which is also treated as set of words, contains more “relevant-words” than “irrelevant-words”, and vice versa.

The method:

**Initialization**

1. For each article \( a \in D \), let \( W(a) \) be the set of words in the abstract of \( a \).
2. Let \( R \subseteq D \), be the set of relevant articles and \( \overline{R} = D - R \), the set of irrelevant articles.
3. Let \( L \subseteq D \), be the training set and \( T \subseteq D - L \) be the test set.

**Training (set-size \( s \))**

1. Let \( L_W = \bigcup_{a \in L} W(a) \).
2. For each word \( w \in L_W \):
   
   - (a) \( \text{Articles}[w] = \{ a | a \in L, w \in W(a) \} \).
   - (b) \( \rho(w) = |\text{Articles}[w] \cap R| / |\text{Articles}[w] \cap \overline{R}| \).
3. Let \( R_{\text{max}}(\rho) \) = the \( s \) words \( w \) of \( L_W \) whose value (\( \rho(w) \)) is the largest.
4. Let \( R_{\text{min}}(\rho) \) = the \( s \) words \( w \) of \( L_W \) whose value (\( \rho(w) \)) is the smallest.

**Testing (certainly-parameter \( \lambda \in [0, 1] \))**

For each article \( a \in T \):

1. \( \text{diff} = |W(a) \cap R_{\text{max}}(\rho)| - |W(a) \cap R_{\text{min}}(\rho)| \)
2. \( \text{max} = \text{MAX}\{|W(a) \cap R_{\text{max}}(\rho)|, |W(a) \cap R_{\text{min}}(\rho)|\} \)
3. If \( (\frac{\text{diff}}{\text{max}} \leq \lambda) \) Claim inconclusiveness (if \( \lambda = 0 \) then inconclusiveness will be claimed only when \( \text{diff} = 0 \))
4. Else If \( \text{diff} > 0 \) the concept is relevant
5. Else If \( \text{diff} < 0 \) the concept is not relevant

**Evaluation**
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.076</td>
<td>0.97</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>0.3</td>
<td>0.076</td>
<td>0.97</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>0.4</td>
<td>0.083</td>
<td>0.98</td>
<td>0.96</td>
<td>0.97</td>
</tr>
<tr>
<td>0.5</td>
<td>0.11</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td>0.6</td>
<td>0.11</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td>0.7</td>
<td>0.134</td>
<td>0.98</td>
<td>0.99</td>
<td>0.98</td>
</tr>
<tr>
<td>0.8</td>
<td>0.159</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>0.9</td>
<td>0.183</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Table 4.6: The results of “using a self developed supervised learning model”. For each certainly-parameter (λ) the table contains the values for unclassified, recall, precision and F-score.

- We tested our method on 270 articles, out of which 135 were relevant to the subject and the rest, 135 articles, were irrelevant.
- In order to establish statistical significant results, 5-fold-cross-validation method was performed.
- The set-size \( s \), which is the size of the “relevant words” set and the “irrelevant words” set, which were created during the training process, was set to 100. Other sizes were examined but yielded worse results.
- The certainly-parameter \( \lambda \) was examined for several values.

Additionally, we wanted to gain some perspective about the coverage, and to try to estimate the general recall. In contrast to the recall values which are presented in Table 4.6, and only take into account the 290 tested articles from \( D \), the size of Wikipedia did not allow us to accurately calculate the recall of this method (from \( W \)), and thus we conducted an additional experiment to estimate it.

**Results**

Table 4.6 summarizes the results for this method. Note that as the value of \( \lambda \) and the F-score grew, the percentage of the unclassified articles also grew, and exceeded 18%. These results are better than the results which presented in section 4.2.4.1 (“using SVM as supervised learning model”). We think that one of the main reasons for that, is the fact that this algorithm has the ability to declare “unclassified” and to filter out concepts, whose classifications are not significant, according to the value of \( \lambda \).

The additional experiment included accumulation of the 38 most relevant collocations from the collocations found in section 4.1, and manually checked if they appear in the concepts set we developed in this section. Meaning, there is an article, which was classified, in this section, as relevant, and its title is the collocation. We tested only
Table 4.7: This table compares between the best results achieved from each method.

38 collocations because we took only the most relevant expressions from section 4.1, and ignored expressions that were irrelevant or uncertain. We found that most of the collocations (90%) were found with this method, 5% were not found, and 5% are not a Wikipedia’s concept at all. These results confirmed the assumption, that most of the relevant concepts are indeed in $D$, and that our coverage is satisfying.

### 4.3 Results summary

Table 4.7 compares between the best results achieved from each method. There is no doubt that the third method (“Expanding the information and adding a learning ability”) produced significantly better results (higher F-score) than the other two. However, it is really important to note that this improvement came with a price. As opposed to the previous methods, this method required a significant effort on behalf of the user – the need to manually tag the learning set, and therefore, the final result is affected from the quality of tagging.
Chapter 5

Finding the relationships between the domain’s basic concepts in the source-language

This is the second and final step in building the ontology in the source language. The goal of this step was to find the connections (edges) between the concepts (nodes), that were obtained in the previous step. For this purpose, we decided to rely on a foundation ontology. The general idea was to find the concepts in the foundation-ontology, and then to derive the appropriate relationships between them.

5.1 Using the “WordNet” foundation-ontology

We tried to use this ontology in order to deduce the relationships between the concepts we accumulated in the previous task, and to create a full ontology. We came across several problems when trying to use this ontology:

- Apart for a small number of exceptions, WordNet does not contain proper names. So, the insertion of names to the ontology had to be made separately.

- Many relevant concepts are missing in WordNet. Out of the 183 manually-classified relevant Wikipedia’s titles, only 18 were found in WordNet. A large part of the remaining 165 articles were proper names, but there were also a lot of relevant collocations we expected to find, for example “drug diversion”.

- WordNet misses many connections. For example WordNet does not contain a connection between “organized-crime” and “crime”. This lack of relevant and important connections in WordNet made it impossible to create a reasonable domain ontology based on it.

When taking into account these three problems, the best action was to use another ontology.
5.2 Using the “Wikipedia” database

As mentioned above, Wikipedia’s database is, in fact, a DAG. If we refer to the nodes (articles) of this DAG as concepts, and to the connections between these nodes (via the categories) as relations, we can view this DAG as an ontology. Wikipedia’s large amount of concepts—especially named entities, along with more accurate and various connections between them (compared to WordNet), made it more suited to our purpose.

5.2.1 Naïve algorithms

There are two naïve algorithms for this problem, which have to be discussed before the describing the algorithm we developed:

1. Using an entire sub-part of Wikipedia – The usage of an entire sub-part of Wikipedia as is, such as the sub-DAG of “organized crime”, cannot be successfully implemented as a method for generating the ontology. As we saw in the previous task, a sub-DAG in Wikipedia contains many irrelevant concepts, which might contaminate the results. Thus, using this sub-DAG as is, yields an ontology with many irrelevant concepts and connections. In order to overcome this problem, the best solution is to find the connections between the concepts that were already found in the previous task. The naïve algorithm, which extracts the relations between these concepts from Wikipedia, does not work in this case. Removing the irrelevant concepts from the domain sub-DAG creates a disconnected graph, with which this naïve algorithm cannot deal successfully. Thus, in case that there are irrelevant concepts inside the sub-DAG, the naïve algorithm fails completely. This problem is critical, and is a good reason to abandon this naïve algorithm.

2. Choose the closest ancestor as the connection – In this naïve algorithm the connection between two concepts is represented by the common, closest, ancestor of these concepts. The problem with this algorithm is that there are many cases in which the closest ancestor is not the obvious option, and choosing it does not always yield the best result. Figure 5.1 demonstrates a sub-DAG derived from $D$. The immediate connection that can be concluded from this sub-DAG is the category “American mobsters of Sicilian descent”. This is a correct connection, but maybe not the best one. A deeper inspection of this graph yields another connection between these two concepts: the category: “Bosses of the Five Families”, which, although is a further away connection, is more accurate and specific. There are many more American mobsters of Sicilian descent, than bosses of the five families, which are all Sicilian descent mobsters.

Another problem occurs when there are two articles, which have two shared ancestors at the same distance, and the method to choose between them should be defined. In Section 5.2.2 we describe an experiment which compares the algorithm
we developed and this naïve algorithm, which chooses the closest ancestor (if there are two ancestors at the same distance, one of them will be chosen).

5.2.2 BFS-based algorithm

The goal in this step was to find the connections, within Wikipedia, between the concepts found in the previous task. For this purpose, we developed a method which finds the “best connection” between the concepts in the Wikipedia database in the way described below. We created the sub-DAG $\overrightarrow{D}$ (i.e., the edges of the reversed DAG lead from leaves of $D$ to $D$’s root), by reversing the directions of the edges of the sub-DAG $D$, and performed a depth-limited BFS (breadth-first search) on each concept (leaves in $D$ and roots in $\overrightarrow{D}$). The depth of the algorithm defined in the initialization point. For each two concepts, we looked at the BFS-tree, found all the intersections, and chose the best one, based on two criteria:

- Number of children – If their number is large, the category is wider and worse.
- Depth (distance from root) – If this number is large, the category is more specific and therefore better.

As mentioned above, we treated only article-category and category-category edges, so the connections between the basic concepts (articles) contained only category concepts. Since all the basic concepts are in the domain, and moreover, they are all in $D$, we concluded that there is a connection between any two basic concepts, which is entirely contained in $D$. Thus we limited our exploration to $D$.

The method is iterative – at each step we performed a single BFS-step for each concept and saved, for this concept, the set of new categories encountered during this step. Then, for each concept, we checked if one of these new categories is already in other concepts’ sets. If there is such concept, a new connection between these two concepts was found. Note, a connection-category which was $n_1$-edges away from the first concepts, and $n_2$-edges away from the second, was found in $\max\{n_1, n_2\}$ steps. When a new connection was found, it was compared to the previous best connection that was found before (if there was one), by the criteria mentioned before, and the best one is retained.

The method can be demonstrated with the sub-DAG presented in Figure 5.1. During the process, BFS is performed on the two concepts in this sub-DAG: “Tommy Lucchese” and “Vincenzo Terranov”. At the first iteration the algorithm finds the categories “Bosses of the Genovese crime family” and “American mobsters of Sicilian descent” for “Vincenzo Terranov”, and the categories “Bosses of the Lucchese crime family” and “American mobsters of Sicilian descent” for “Tommy Lucchese”. Thus, a mutual category is found for these two concepts. Since these two concepts do not have another connection yet, this category is defined as their connection. In the second iteration, the category “Bosses
Figure 5.1: Example for a sub-DAG of D. The concepts in the quadrilaterals are the categories, and the concepts in the circles are the articles. The connection represented by an arrow is "part-of" (as described in section 3.3).

of the Five Families" is found for both concepts (for "Vincenzo Terranov" through "Bosses of the Genovese crime family", and for "Tommy Lucchese" through "Bosses of the Lucchese crime family"). Since another connection is found at the second iteration for both concepts, this connection should be compared to the existing connection, and only the best one is retained.

Algorithm (number-of-iterations $i$, sub-trees-parameter $a$, depth-parameter $b$)

- Create the sub-DAG $\overrightarrow{D}$ by reversing the directions of the edges of the sub-DAG $D$.
- For each concept create a single node tree, with the concept as its root.
- For $i$ iterations, for each concept $c$:
  1. Perform a single BFS step of the concepts trees (expand the graph of each concept) within $\overrightarrow{D}$, and add, the newly discovered concepts to the tree of $c$. (Ignore concepts that already belong to the tree).
  2. If, as a result of this step, the BFS trees of two concepts intersect:
     For each intersection node:
     (a) If no previous connection was found between those concepts:
i. Retain the intersection node as the “connection concept”

(b) Else:
   i. Let $n_1$ be the previous “connection concept”
   ii. Let $n_2$ be the new “connection concept”
   iii. If (Is-better ($n_1$, $n_2$, a, b)):
        Remove the old connection
        Retain the new “connection concept”.

Is-better (current-node $n_1$, new-node $n_2$, sub-trees-parameter a, depth-parameter b)

1. Let $c$ be the maximum number of children of one node, from all the nodes in $W$ (Wikipedia’s DAG)
2. Let $d$ be the maximum depth in $W$ (the maximum distance from node to the root)
3. For $i \in \{1, 2\}$:
   (a) Let $c_i$ be the number of children of $n_i$.
   (b) Let $d_i$ be the depth of $n_i$.
   (c) Let $v_i$ be $(\frac{c_i}{c} \times a) + (1 - (\frac{d_i}{d} \times b))$.
4. if $v_1 > v_2$ return true
5. else return false

Evaluation
We claim, based on the assumption made earlier and on the final results, that the algorithm finds a connection between every pair of concepts, and that these connections are in the domain scope. On the other hand, it is difficult to measure the quality of the found ontology. When trying to do so, several questions came up:

- What is the definition of quality ontology?
- How to compare two ontologies?
- To which ontologies it will be valuable to compare?

We claim that the suggested algorithm is better than the naïve algorithms described above. It is clear why the suggested algorithm is better than the first naïve algorithm, which does not work at all. So, we had to provide justification that the suggested algorithm is better than the second naïve algorithm. In order to do that, we performed an experiment, in which both algorithms were run on 200 pairs of concepts, and the connections found between them were compared manually.
Table 5.1: This table compares between the naïve algorithm and the suggested algorithm.

<table>
<thead>
<tr>
<th>Compared pairs</th>
<th>Same results</th>
<th>Better result in the naïve algorithm</th>
<th>Better result in the suggested algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>114</td>
<td>0</td>
<td>86</td>
</tr>
</tbody>
</table>

Table 5.1 compares between the naïve algorithm and the suggested algorithm. Although for most of the pairs, the naïve algorithm and the suggested algorithm provided the same connection, for 32% of the pairs the provided connection was different. In these cases, the connections provided by the suggested algorithm were better.
Chapter 6

Creating the corresponding target-language ontology

Our task in this chapter was to build the bilingual-domain-ontology, by developing the target-language (TL) ontology based on the source-language (SL) ontology we created in the previous sections. In this task it was proved that a corresponding Hebrew-ontology can be developed from the English-ontology based on Wikipedia. Wikipedia is used for many cross-language tasks, such as information retrieval. There were several works on this subject such as the following: Potthast, Stein and Anderka [20] introduced A Wikipedia-Based Multilingual Retrieval Model. Sorg and Cimiano [25] described an approach, which indexes documents with respect to explicit interlingual concepts. Nguyen, Overwijk, Hauff, Trieschnigg, Hiemstra and de Jong [19] presented WikiTranslate, a system which performs query translation for cross-lingual information retrieval (CLIR) using only Wikipedia to obtain translations. The works we encountered on this subject are concentrated on developing a method for aspects of information retrieval, based on a single input. In our work, we extended these approaches to analyze a complete domain, and to develop a whole corresponding ontology in the TL.

The resulting Hebrew-ontology of this section does not contain the same exact concepts and relations as the English-ontology. The main purpose of this task was to develop a multilingual domain-ontology, which has interlinks between the TL concepts and their corresponding SL concepts, if they exist. As a result, not every concept in one language has a corresponding concept in the other language. During this task, we relied on Wikipedia’s structure, which does not contain the same concepts for each two languages. It is easy to understand why the fact that the SL Wikipedia is much bigger and wider than the TL Wikipedia, effects the number of relevant SL concepts which are not exist in the TL Wikipedia. But, surprisingly, there are many relevant concepts which exist only in the TL Wikipedia – most of them represent concepts
that are relevant mostly to the TL culture. For example, in our examined domain and languages, names of Israelis criminals are better represented in the Hebrew Wikipedia, in comparison to the English Wikipedia.

The task of filling the missing concepts in the two ontologies, in order to create a completely corresponding multilingual domain ontology, involves very hard challenges as translating names and concepts. For example, in the experiment where SL is English, and TL is Hebrew, the word “Mark” can be translated as a name, or as a word (sign). Another example is the polysemous “Lean” (thin and rely), and each one of them has different represented word in Hebrew. These challenges are not in the scope of this work, so in this chapter we concentrated on methods which deal with the task of creating a TL ontology, based on the SL ontology, and the task of creating the relevant interlinks between them.

We developed three methods for this purpose – The first two were unsuccessful, but the third, successful, method relied on them. The dataset for all the three methods was the 145 articles set, that was manually classified as relevant to the domain in discussion (Section 3.3).

6.1 Using Wikipedia’s interlinks to get the target-language Concepts

In order to get the TL concepts we used Wikipedia’s interlinks between almost any two languages, in this case English to Hebrew. We expected to find such connection for most of the 145 topics, but actually found a connection for only 27 (19%). The reason for the absence of the remaining 81%, is because the TL Wikipedia is much smaller than the SL Wikipedia. This result demonstrates why it is a very difficult task to build an ontology in resource restricted languages. Denote the set of these 27 TL topics by \( H \). Naturally, the false-positive was 0, since all the concepts which were relevant in the SL, continued to be relevant in their TL version. On the other hand, the false-negative value was substantial – Many articles that appear in the TL Wikipedia do not appear in the SL one. In our example domain and languages, many articles deal with Israeli mobsters. Restricting our attention to the TL articles, that have an related article in the SL, we would miss a lot of very relevant concepts in the TL, which appear only in the TL Wikipedia. Since we decided to concentrate just on Wikipedia, we had no tools to deal with the 81% of the SL topics that do not have a corresponding TL one. Notice that the proportion in the not-found topics between proper-names and not-proper-names was equal to the proportion between these groups in the 145 examined articles. Meaning, that we cannot handle relatively easily one of these groups. Instead, we concentrated on how to reduce the false-negative value by adding to the TL ontology the concepts which are unique to TL Wikipedia.
6.2 Extending the target-language concepts by their target-language categories

In order to reduce the false-negative value, we tried to find additional relevant concepts in the TL Wikipedia DAG. The assumption was that the relevant concepts that were not found are probably DAG-neighbours of the ones already found. So, creating a set $C$ of the TL categories of all the relevant TL concepts, allowed us to determine the relevance of each concept. Thus, the algorithm was as following:

Input: A set $H$ of relevant TL articles
Output: A set of additional relevant TL articles
$C = \{c|c \text{ is a TL category of at least one TL article } a \in H\}$
return $\{b \in \text{TL Wikipedia} | b \text{ has at least one ancestor in } C\}$

The result for this method was poor – The vast majority of the retrieved articles were not relevant. The reason for that is, as shown before, a large part of the categories of a relevant concept in Wikipedia are irrelevant. (For example: year of birth, origin and more.) Thus, when looking at the other children of such categories, most of them are irrelevant.

6.3 Extending the target-language concepts inside the target-language domain sub-DAG

In order to implement the ideas presented in the previous method with a smaller value of false-positive, we decided to concentrate only on the TL domain sub-DAG (in our example: Hebrew organized-crime” sub-DAG) called $HD$, which, similar to the SL counterpart (Section 3.3), is the sub-DAG, rooted at the TL node corresponding to domain name (in our example “Category: Organized crime”). The algorithm is:

Input: A set $H$ of relevant TL articles
Output: A set of additional relevant TL articles
$C = \{c|c \text{ is a TL category of at least one } a \in H\}$
Let $R$ be $C \cap HD$
return $\{b \in \text{TL Wikipedia} | b \text{ has at least one ancestor in } R\}$

Evaluation and conclusions
After executing the algorithm we got 43 topics in Hebrew. We manually classified them, with the same criteria used for the English articles, and got no false positives. Hence, all of the articles are relevant. Calculating the false-negatives was more complicated.
There were relevant article topics in $HD$ which were relevant and were not classified by the algorithm as relevant. Notice that, in order to find the relevant Hebrew concepts, we used only 145 English relevant concepts, and not all of the relevant English concepts. We inspected the categories of the Hebrew concepts which were mistakenly classified by the algorithm as irrelevant, and manually proved that each one of these categories includes at least one article which has an English corresponding relevant article. Therefore, the conclusion is that if we had used all of the English relevant articles during the translation process, we would get no false-negative inside $HD$. This conclusion cannot be proved, because of the size of Wikipedia in general and the size of $D$ in particular.
Chapter 7

Applying the method on a different domain

We claim that the method presented in the previous chapters is general, and therefore can be implemented for other domains as well. The domain “Organized Crime” was chosen as an example in order to perform experiments and evaluate the method and all its internal tasks. In order to prove the generality of the method, we executed the method on another domain – “vehicles”. First, we had to prove that the naïve approach of extracting the sub-DAG of “vehicles” from Wikipedia results in a poor ontology. For this purpose, we accumulated a database containing 240 articles, which were collected randomly from the set of the concepts in the domain of “vehicles” in Wikipedia, according to the technique presented in Section 3.3. Out of these articles, only 90 articles (37.5%) were relevant. The remaining 150 articles were irrelevant and concentrated on cities, movie stars, etc.. This statistic proves that the sub-DAG of “vehicles” in Wikipedia contains irrelevant concepts as well. Thus, for this domain is also impossible to simply extract the suitable sub-DAG, and more complicated algorithm, like the one we proposed, is required.

For each of the method’s tasks, we ran the algorithm on a balanced database, which was derived from the 240-articles-database mentioned above. The examined database contained 180 articles, of which half (90) were relevant to the subject, and the rest were not.

7.1 Accumulating the domain’s basic concepts in the source-language

In this experiment we evaluated the method, which was described in section 4.2.4.2, on the balanced database described above, which contained 180 articles. We performed a 5-fold-cross-validation method, with the parameters \( s \) (the size of the “relevant/irrelevant words” set) set to 100, and \( \lambda \) (certainty-parameter) set 0.8. In Section 4.2.4.2 we tested
Table 7.1: The results of performing the algorithm for “Accumulating the domain’s basic concepts in the source-language”, for both domains

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicles</td>
<td>0.8</td>
<td>0.166</td>
<td>1</td>
<td>0.95</td>
<td>0.98</td>
</tr>
<tr>
<td>Organized Crime</td>
<td>0.8</td>
<td>0.159</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
</tbody>
</table>

several values of λ, and there were not large differences in the results. Thus, in this experiment we concentrated on one value of λ which was proven to gain the best result in the section under discussion.

Results and conclusions
Table 7.1 summarizes the results for this experiment and compares it to the corresponding results in Section 4.2.4.2. As can be seen in the tables, the results for the domain “Organized Crime” are slightly better than the results for the domain “Vehicles”, but both are very good, and therefore proved that this algorithm works for at least another domain.

7.2 Finding the relationships between the domain’s basic concepts in the source-language

In this section we implemented the algorithm described in Section 5.2.2. The algorithm described there proved to be better than the naïve algorithm, and therefore in this experiment we did not compare it to the naïve algorithm. Instead, we looked at the found connection, and ensured that these connection make sense.

Examples

- The connection between “Mazda RX-7” and “Rambler Marlin” is the category “Rear wheel drive vehicles”
- The connection between “Mazda RX-7” and “Cyclekart” is the category “Auto racing”
- The connection between “Avro Canada C102 Jetliner” and “General Aircraft Cagnet” is the category “Low wing aircraft”
- The connection between “Ford Festiva” and “Ford Crown Victoria Police Interceptor” is the category “Ford vehicles”

All these connections are reasonable, and thus we resolve that this method works well for this domain.
7.3 Creating the corresponding target-language ontology

In order to examine the third task, we performed the algorithms described in Chapter 6. 6.1 and 6.3.

First we executed the algorithm described in Section 6.1. Out of the 90 relevant concepts in the SL (English), only 3 (0.3%) were found in the TL (Hebrew). Then, we executed the algorithm described in Section 6.3, and tried to find more relevant articles. In this experiment 40 additional articles were found.

In Section 6.3 we claimed that the fact that in these experiments we do not find all of the relevant concepts is due to the size of the experiment. We tried to find the relevant concepts in the TL based on only a small set of relevant concepts in the SL. If we were able to execute this experiment on all of the relevant concepts in the SL, we would probably find all the concepts in the TL. This claim is relevant here as well, and execution of the method on the domain “vehicles” provides similar results to the domain “Organized Crime”.

7.4 Conclusions

As can be seen above, execution of the 3 algorithms of the method on different domains provides similar results. The obvious conclusion is that the entire method is general, and probably can be implemented on more different domains as well.
Chapter 8

Summary, Conclusion and Future work

8.1 Summary

In this work we created a multilingual domain ontology in English and Hebrew in the “organized crime” domain, and examined it in the “Vehicles” domain as well. The research was divided into three sub-tasks.

The first was concerned with gathering the most relevant concepts to the domain. For this task we examined two data sources: contrasting corpora and Wikipedia. While the contrasting corpora provided insufficient results, the Wikipedia results were satisfying, and its F-score is 0.99.

The second sub-task found the relations between the concepts which were found in the first sub-task, and constructed an ontology out of them. In this sub-task both WordNet and Wikipedia were examined as foundation-ontologies to rely on. In contrast to our assumptions, WordNet lacked many important concepts and connections. Fortunately, the Wikipedia ontology turned out to be suitable.

In the third sub-task the ontology which was created in the previous tasks, was partially translated into Hebrew, and a bilingual–ontology was created. For this task we relied on both the English and the Hebrew ontologies, and provided a result based only on these ontologies.

8.2 Conclusions

- Wikipedia is a rich and broad foundation ontology, and it can be used as a reliable source for all the steps in the ontology building process: the concepts gathering and the relations creating. As Wikipedia will expend over time, we estimate that the quality of the resulting ontologies will also improve.

- In contrast to our expectations, WordNet turned out to be unsuitable for our
purposes. Although it is a versatile ontology, which contains concepts from many different subjects, when examining the coverage of WordNet for one specific field, it failed to provide sufficient results.

- The contrasting corpora approach also did not provide satisfying results. It seems that for such a task, in which the domain is not well defined, this database is not suitable.

- We gave a proof of concept for developing a multilingual domain-ontology based only on foundation-ontology. Based on this proof, other domain ontologies, in other languages, can be built for different purposes, in different fields.

8.3 Future work

- Enrich the Hebrew ontology—As mentioned in Section 6.1, the developed Hebrew ontology lacks many relevant concepts due to the size of the Hebrew Wikipedia. Thus, an additional effort needs to be invested in order to translate the remaining concepts, using other tools and methodologies. This task will compliment our results, and allow us to construct a more comprehensive ontology.

- Examine additional databases and foundation ontologies—In this work we examined the currently, most famous and widespread ontologies, along with one database. Additional ontologies and databases have to be examined as well, and compared to those we chose.
Bibliography


42


1. The research was conducted in the English and Hebrew languages. The English and Hebrew ontologies were created to include concepts relevant to the topic “organized crime.”

2. The research involved the use of related concepts in Wikipedia to achieve relevant terms in the Hebrew language. In this manner, the Hebrew terms were translated using the relationships between articles in the Wikipedia articles, which contain the Hebrew language. This is possible because Wikipedia contains over 27% of the relevant terms, which are classified as important for the topic. Furthermore, the authors considered that the Hebrew terms are translated from related terms. The authors decided to maintain this approach because the English language contains many related terms that are not included in the Hebrew language. Therefore, the authors used the automatic translation system to achieve relevant terms in the Hebrew language. Moreover, the authors used related concepts to extend the Hebrew ontology. This is similar to the previous approach, as the authors focused on extending the ontologies in the Hebrew language with related concepts.

3. The research involved the use of related concepts in the Hebrew ontology. The researchers used related concepts to extend the Hebrew ontology with related concepts.

4. The research involved the use of related concepts in the Hebrew ontology. The researchers used related concepts to extend the Hebrew ontology with related concepts.
The extraction process for the category matched the query group (for example, "movies") with the results obtained. The method has proven itself and the results have been published in other. In our opinion, the gap between theory and practical knowledge is often close to a distortion.

The foundation is based on the summary of each article in Wikipedia, for the most part. Initially, it divided the articles into two groups: a relevant group and an irrelevant group. After that, with the help of the two groups, the articles were examined. If the summary of an article contains more words from the relevant group, it was classified as relevant. If the summary of an article contains more words from the irrelevant group, it was classified as irrelevant. If the words in the summary of an article fall between the two groups, it was classified as neutral.

Conclusions and recommendations:

Precision = Recall = True-positive / (True-positive + false-negative) = 0.99
F-score = 0.99 True-positive / (True-positive + false-positive) = 0.99

In order to create an ontology, a summary of the relations between the relevant terms was found. The idea in general would be to add the relevant terms to the ontology, where every term relates to the same ontology that contains concepts in various fields and is not specific to any particular topic. We developed several methods, each of which is based on an ontology that is not based on any specific factors

WordNet

Ontology ontology is a collection of concepts. An ontology is a representation of knowledge in the form of a network of interrelated concepts. In this way, it can be used to create an ontology.

WordNet

In conclusion, we recommend using the WordNet ontology, as it was found to have the best results. The number of entailments is less in the WordNet ontology than in the original WordNet ontology.

In the first stage, we created, in accordance with the WordNet ontology, the corresponding ontology in the Hebrew language.

Technion - Computer Science Department - M.Sc. Thesis  MSC-2016-01 - 2016
The relevant concepts will be found in the article. In order to estimate the methods and proceed, we will collect a subset from articles from the "organized crime" subgraph of Wikipedia, and then select a subset of articles from Wikipedia. Each subset will be relevant to the concept "organized crime," as well as 50% relevant to the concept. Further steps will be taken with the subset of relevant articles.

The tasks will consist of the following three stages:

1. The task includes three sub-tasks, each of which presents a new challenge and serves as the basis for the sub-tasks.
2. The sub-tasks are:
   - Collecting the key concepts central to the topic and creating an ontology in the Hebrew language.
   - Translating the main concepts, using statistical methods, which will be used to build an ontology in the Hebrew language.
   - The task consists of the following steps: A bilingual corpus of Wikipedia, which will be used to automatically collect a subset of relevant expressions in the field of investigation. In addition, we will use the following measures: TF-IDF, PMI.
3. The measures are combined using three statistical measures and return a score for the degree of fit. In order to estimate the method and the statistical measures, the expressions that received the highest score in the subset were classified as true positive, whereas the other measures did not provide any results. The classification was based on the categories of the method: 2 articles are relevant if they are included in the category. An article is classified as relevant if it is included in any of the categories. The highest F-score value was obtained, given by the variation in the number of categories of non-relevant articles such as birth year, origin, etc.

The idea of the method is to remove articles from the "organized crime" subgraph. The method works as follows: 1. Collect an article classified as relevant in the "organized crime" subgraph. 2. Classify the article as relevant if it is included in the category. 3. Collect an article classified as relevant if it is included in any of the categories. The highest F-score value was obtained, given by the variation in the number of categories of non-relevant articles such as birth year, origin, etc.

Technion - Computer Science Department - M.Sc. Thesis  MSC-2016-01 - 2016
תקציר

הנה מבנה ארגון מידע, אשר מתאר את עולמו המושגי ואת הקישורים ביניהם. המטרה היא לבנות עולמות מושג מסוימים, פיתוחים ב/embedding וโบราณים באמצעות אונטולוגיות ליציאתם על יד@Api. המודל ולשון שלsteen, המדגים את הקשרים ביניהם, יוצר ניסיון של הראות קלות חידוש העתקה. מסר למודל החוזק בהיב violated, מייצרת אונטולוגיות דלילות, המyyyyMMdd את המושג

העדכון של התחום, כאשר בחרים, מייצרים אונטולוגיות שלידן, המ HttpResponseMessage על יד(Api

בעשורים האחרונים, השימוש באונטולוגיות לתיאור עולמות מושג התגבר, ומודל זה משמש גם מודל לתורת ת יהודי עלייה, יוצר אונטולוגיות

בשופט יקשי ביניהן, התוכן שבחר שלב פותח יבלוב והוי "פשע וחרות"على יד(Api

השונים: כלכל, אידיאולוגי, טרור(ו') etc. הנושא שבחר למחזור בחתת האונטולוגיות בבלוב והוי "ילח חכימה". השפה הנוספת שشرع בהברנה הוא יליו הנושא עבור הנושא שערוי, על מנטเหนית והחזרה

הממכרות, האטרקציה חוכל לשלוש ת"שת"ים, אשר ייגון ה医药פ

מוצרי מידע

בנולי העבדה נהגה שלושה שיטות המאגרים באמצעות ב机械设备 אונטולוגיות:

- קורופוסי מוגנים: הקורופוס הרחואני מכיל 800 אמנים, ראש חפץ ברווז אנוטוט את אונטוטית ואקטיבית הנדנוד לגישיה שלושה, בודק הקורופוס השיא מכיל 8304 אמנים ברווז אנוטוטית, והוור פופולות בבלוב

- WordNet – מטיל אונליין מוכנה אשר פותח באוניברסיטאות פורנרט, מחמירות, והם בבלוב

- האוניליט בבלוב שמסתייע דומע, מכיל את הקישורים בין בקצות

יקפדייה- הקפידייה היא אנציקלופדיה והקורות שלップ, אשר מכילה מטיל בר של אוניליט

- 4,252,811 ערכו באוניליט ביאן (2013) ברומן הבור שלצעים. היקפידייה מתכתי הולכת והתמכה 추진ית. יחידי הלעיים שלפז הקפידייה, בבלוב

השונים בבלוב שמקני מתכתי היקפידייה, כה היקפידייה

- אשר הור מלקאנה מחיק, כה אוניליטית

- מצפת בברך – מרחבי אוניליטית: ביקפידייה יש של צופים: מפרים, תקופות מתכתי פסכים ומושג מייסדים, הקישורים שלושה ואילו שבל החיזוק הנוספים בבלוב

- מרחבי אוניליטית ישר מנשיכים באנטולוגיות

- קשוחות בברך – קשורים באנטולוגיות: מתחים של צופים קשוחות ביקפידייה: קפידיית-קפידיית ר

- קפידיית מאמץ קשוחות על יד_Api. המחסנית בברך

- מצפת בברך – קשורים באנטולוגיות: מתכתי של צופים קשוחות ביקפידייה: קפידיית-קפידיית ר

- קפידיית מאמץ קשוחות על יד_Api. המחסנית בברך

- מצפת בברך – קשורים באנטולוגיות: מתכתי של צופים קשוחות ביקפידייה: קפידיית-קפידיית ר

- קפידיית מאמץ קשוחות על יד_Api. המחסנית בברך

- מצפת בברך – קשורים באנטולוגיות: מתכתי של צופים קשוחות ביקפידייה: קפידיית-קפידיית ר

- קפידיית מאמץ קשוחות על יד_Api. המחסנית בברך

- מצפת בברך – קשורים באנטולוגיות: מתכתי של צופים קשוחות ביקפידייה: קפידיית-קפידיית ר

- קפידיית מאמץ קשוחות על יד_Api. המחסנית בברך

- מצפת בברך – קשורים באנטולוגיות: מתכתי של צופים קשוחות ביקפידייה: קפידיית-קפידיית ר

- קפידיית מאמץ קשוחות על יד_Api. המחסנית בברך

- מצפת בברך – קשורים באנטולוגיות: מתכתי של צופים קשוחות ביקפידייה: קפידיית-קפידיית ר

- קפידיית מאמץ קשוחות על יד_Api. המחסנית בברך

- מצפת בברך – קשורים באנטולוגיות: מתכתי של צופים קשוחות ביקפידייה: קפידיית-קפידיית ר

- קפידיית מאמץ קשוחות על יד_Api. המחסנית בברך

- מצפת בברך – קשורים באנטולוגיות: מתכתי של צופים קשוחות ביקפידייה: קפידיית-קפידיית ר

- קפידיית מאמץ קשוחות על יד_Api. המחסנית בברך

- מצפת בברך – קשורים באנטולוגיות: מתכתי של צופים קשוחות ביקפידייה: קפידיית-קפידיית ר

- קפידיית מאמץ קשוחות על יד_Api. המחסנית בברך

- מצפת בברך – קשורים באנטולוגיות: מתכתי של צופים קשוחות ביקפיד
המחקר נעשתה בהנחיית פרופסור אלון אייתי, בפקולטה למדעי המחשב.

תודה

ברצוני להודות לפרוור, אלון אייתי על העזרה והתמיכה במחדק立ちור בשניים. שליט, ברצוני להודות על้า לי איננה והרי, לאחתי, אלנה הורין, לאחתי, מיכל,لسבתו, נלי קפלבך, לבלי י comparer וחברי עליה, ברצוני להודות על תופעת אילנות השועה והאמונה בבלי כל השציים והללים. שילושי, ברצוני להודות על אלי תמיכת, בדefdיה מדיה ליונתן צ'רצקסי, שהכניס אור ומושעת לחיי. ואחרון חביב, ברצוני להודות על כל התמיכח תמיכה והบังורה בחרשה של, מיכאל צ'רצקסי. למדתי אתט ויתר ושאול על מסע להראות.

אני מודה לטכנון על התמיכה הכפיפה הנדיבת והשתלמות.
כיزن לבנות אונטולוגיות תחומים רב לשוניים

חיבור על מחקר

ሰשם ملييح חלקי של הדרישות לקבלה התואר
מיניסטר להדיעו במדעי המחשב

גיון חירוזן אָרֶנסקי

hppה יותם הטכניון - מכון טכנולוגי לישראל
שבט התשע”ח היפקט
דצמבר 2015
כיצד לבנות אונטולוגיות תחומי רבי לשוניים

גיורם הירשון ארקנסקי