Computational Linguistics and History of Science

John Kizito, Ismail Fahmi, Erik Tjong Kim Sang,
John Nerbonne, and Gosse Bouma

Abstract

We present computational linguistics techniques which can help researchers dealing with the increasing amount of available digital text, focusing on potential use in the history of science. We examine two tasks: automatically extracting terms from documents and identifying relations between the terms. We show that the two tasks can be performed reliably and fast, and that new terms and relations can be identified automatically.

1 Introduction

In the recent decade, there has been a large increase in the amount of digital text available. These new data collections offer great opportunities for researchers, and this paper will sketch how INFORMATION EXTRACTION (IE) proceeds on the basis of large reserves of text.

COMPUTATIONAL LINGUISTICS (CL) has sought to extract information automatically from increasingly large collections of texts for some time now. For example, in a competition in 2007, participants were required to build systems that process up to a million documents and from these documents, extract answers to arbitrary questions. Twenty-two systems participated in the task [Giampiccolo et al., 2007].

Typically, CL work processes articles from newspapers and encyclopedias. Recently there has also been an increasing amount of work with documents from the medical domain, and we shall focus on this domain in the present article as its texts provide similar challenges as those of interest for historians of science. In fact, there has been little work applying CL to texts used in other fields such as history and literature.[1] We believe that CL can provide useful contributions to these fields, but we shall not attempt to survey all the uses to which CL techniques might be put in studying history or literature (but see Nerbonne [2007] for remarks on applications of text classification), but focus rather on how information is extracted about particular domains of inquiry.

It is worth noting that a great deal of the CL work is practically motivated by the wish to provide flexible access to textual information and to organize it in ways conducive to automatic reasoning. The fact that there is practical motivation for the work suggests that it will continue, and presumably continue to improve, even if its contribution to current history of science is quite modest. This means that the field may be of strategic value to the HISTORY OF SCIENCE

[1] But see Cardie and Wilkerson [2008] for applications to political science and Hirst and Feiguina [2007] for applications to literary scholarship.

in the longer term. A second consequence of the practical motivation is the wish to quantify progress, and we shall return to this below (2).

This paper presents two examples of CL techniques that we believe could be valuable to researchers who wish to track the content in large text collections: automatic terminology extraction and automatic relation extraction. We advocate exploring a closer cooperation between CL and studies in the history of science, following Dibattista [2003], envisaging the potential value of the ability to track technical vocabulary in a scientific field. This might concern the introduction of novel terms or novel associations among existing terms as indications of innovation; it might mean detecting other changes in technical vocabulary, e.g., frequency changes, as indications of changes in scientific ideas, or at least, changes in scientific attention; or, most ambitiously, it could mean deriving a good deal of information about the ontology of a field automatically, where we understand ontology to comprise the classes of objects studied and the relations attributed among those objects. We present techniques aimed at deriving ontological information from medical texts in the remainder of this article. We hope in this way to stimulate interest in CL techniques as they may be used in studying the history of science.

After this introduction, we deal with terminology extraction in Section 2. We will explain what techniques can be used for this task and look at term variation and term labeling in more detail. In Section 3, we will look at relation extraction. We will show how text patterns for relation extraction can be learned from a few examples and how they can be evaluated. We will conclude in Section 4.

2 Term Detection

This section describes experiments in which terms were extracted automatically from texts. We start by introducing the task and presenting a general description of our system. After this, we present the data and the preprocessing methods used in the experiment. Next, we discuss the extraction techniques used in the experiments as well as their performance. The last sections present two challenges for the extraction process: dealing with term variation and assigning labels to terms.

2.1 Introduction to Term Detection

There are several formulations of the meaning of TERMINOLOGY given in dictionaries and encyclopedias. However, in general, they refer to terminology as a study or a system of TERMS used in a special jargon, discipline, subject, or context [Fahmi, 2008]. Formally, a term is defined as "a designation consisting of one or more words representing a general concept in a special language" [ISO, 2000].

Terms are different from words. The most frequent method of term formation is through combinations of existing lexical elements in particular ways [Sager, 1997]. However, Sager indicates that the formation of complex terms consisting of two or more lexical elements is not always straightforward, since there is no simple linguistic criterion for distinguishing between complex terms (e.g., *high-density disk*) and free-formed phrases (e.g., *high pressure*).

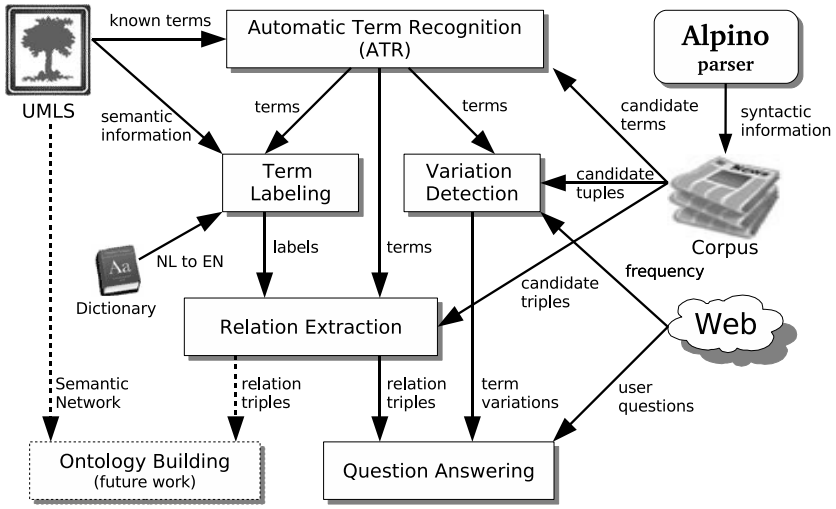

Figure 1: Architecture diagram of the system used for the experiments described in this paper.

Automatic term extraction uses computer programs to identify strings from text that are potential terms [Fahmi, 2008]. This can be done completely automatically or semi-automatically and is referred to as automatic term extraction, semi-automatic term extraction, automatic term recognition, or term extraction. In the rest of this paper, we call this process TERM EXTRACTION.

Automatic extraction of information from text is very important to many applications, such as to information retrieval, question answering, and bootstrapping or extending ontologies for the semantic web. Naturally, it would be gratifying to see that research in pure science, such as the history of science, political science, or literary studies, could benefit from techniques developed for applied purposes.

In our own work on term and relation extraction, we use a modular system which has been described in Figure 1. The system consists of four main processes which will be discussed in this paper: Automatic Term Recognition (ATR, Sections 2.3 and 2.4), Variation Detection (Section 2.5), Term Labeling (Section 2.6), and Relation Extraction (Section 3). Two more applications which will not be discussed here, i.e., Ontology Building and Question Answering. For more information on these two applications, we refer to Fahmi [2008].

In the ATR experiments, medical terms are extracted from sentences using both linguistic and statistical approaches. The input sentences were automatically enriched with a syntactic analysis by the Alpino parser (see Section 2.2). The output of the ATR process is a set of candidate terms ranked by their "termhood", a score assigned by the process based on textual properties. We apply a threshold to select elements with high termhood scores. We then label these terms with medical semantic labels in the Term Labeling module. For this purpose, we label candidates with semantic information from the Semantic Type found in the Unified Medical Language System (UMLS), a collection

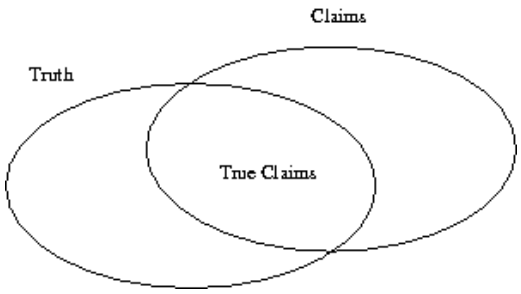

Figure 2: A graphical display of the evaluation of information extraction. The output of a system (Claims) is compared with data annotated by experts (Truth). The results are rated by two scores: precision, the number of true claims divided by the total number of claims, and recall, the number of true claims divided by the total number of true data items.

of lists with definitions of medical terms. Among these terms, there are term pairs which are each other's synonyms. We detect these synonyms as well as term abbreviations in the Variation Detection process (Figure 1). Two terms are marked as related synonyms or related term-abbreviation pairs based on the frequencies of their occurrences in prespecified text patterns in Web texts.

Terms and their semantic labels, together with candidate-relation triples extracted from the parsed sentences, become inputs for the Relation Extraction process (Section 3). To generate relation patterns, we use a subset of the corpus whose sentences have been annotated manually with relation types. Based on these patterns, we extract the candidate-relation triples from the rest of the corpus. For example, we extract relation triples that can be used to generate information tables for a Question Answering application, or extract concept and relation instances for the Ontology Building process.

As we noted in the introduction, a great deal of the CL work on IE is practically motivated. This, together with the complexity of the task, has led to the adoption of concrete measures of success. In the case of IE (and INFORMATION RETRIEVAL) we compare the output of our systems with document collections where the correct results of processing have been identified by experts in the domain under investigation. If we are trying to identify terms in a particular domain, then these will have been marked in the test collection. The list of expertly annotated terms constitutes the "truth", the putative terms we claim to detect are claims, and our success can be measured on the basis of the proportion between the intersection of these two, i.e. the " true claims" on the one hand and the claims but also the entire true set on the other. PRECISION measures how much of what a system has detected, is true according to the experts. Additionally, RECALL measures how much of the expert truths has been found by the system. See Figure 2 for a sketch.

Note that precision and recall are normally inversely related to each other, meaning that by accepting lower recall you can achieve higher precision and vice versa. If, for example, one returns all possible candidates recall will be 100% (and precision infinitesimal). If, on the other hand, very few candidates are proposed, precision is likely to benefit, while recall will suffer.

There are variants of this scheme used in cases where it is infeasible to have test collections annotated completely, e.g. UNINTERPOLATED AVERAGE PRECISION, which computes the precision of a each initial subset of a ranked list of terms and outputs the average of all the precision scores [Manning and Schütze, 1999, 535ff]. In this case we need to specify the size of the ranked list, for example of the first 500 elements and its uninterpolated average precision (see Section 2.4).

2.2 Data and Preprocessing

In this section we describe the foundation of our term extraction work: the text collection and the terminological resources as well as the automatic linguistic analysis used for preprocessing the data. Our text collection has been built from two Dutch medical texts, namely:

1. Elsevier's medical encyclopedia: a medical encyclopedia intended for general audience and containing 379K words.[2]

2. Dutch edition of the Merck Manual: a general-purpose medical handbook intended for professionals and containing 780K words.[3]

The corpora are medical reference books and provide definitions for each term, and for terms describing diseases also their symptoms, causes, diagnosis, and treatment. Consider, for example, a translated article in the Dutch edition of the Merck Manual as shown in Figure 3. This article provides medical information for the disease *acute necrotizing gingivitis*. Some substrings in the article are printed in italic to indicate that they are also terms, for example, *Plaut-Vincent angina*. Our term extraction task is aimed at extracting these kinds of terms automatically.

For the medical domain, two sources of terminological or external knowledge are available, i.e. UMLS (noted above), and a collection of small Dutch terminologies collected from the Internet. Since UMLS contains few Dutch terms, we translate new Dutch terms found in the corpus into English in order to get semantic types from their matching terms in the UMLS. For the term extraction task in this section, we use both of the resources. This is a point where applied research is pleased to exploit additional resources in order to optimize performance. But naturally, one may not always expect to find such resources, especially in connection with historical research. There it will be essential to extract terms exclusively from texts.

2.3 Using Filters for Extracting Terms

The first step in the term extraction process is to find candidate terms in text by looking for context patterns containing specific words or syntactic constructions. These context patterns are called filters. In this section, we compare two linguistic filters, namely a PART-OF-SPEECH (PoS) tag (or WORD CLASS) filter and a syntactic filter.

[2]The encyclopedia was made available to us by Spectrum b.v., and can also be found online at www.kiesbeter.nl/medischeinformatie/

[3]www.merckmanual.nl

ACUTE NECROTIZING GINGIVITIS

Acute necrotizing gingivitis (*Plaut-Vincent agina*, *acute necrotizing ulcerative gingivitis*) is a *painful*, *non-contagious infection* of the *gums* which causes *pain*, *fever* and *fatigue*. This *disorder* is also called *trench mouth*, a term from World War I when many soldiers contracted the *disease* in the trenches. [...]

SYMPTOMS

Often *acute necrotizing gingivitis* start suddenly with *painful gums*, an *uneasy feeling* and *fatigue*. Furthermore, there is *bad breath*. The *gums* are affected and are covered with a gray layer of *dead tissue*. The *gums* start *bleeding* and *eating* and *swallowing* become *painful*. Often the *lymph glands* start to *swell* while the *patient* experience a *light fever*

TREATMENT

The *treatment* starts with a careful but thorough *cleaning* of the*teeth* whilst the *dentist* removes all *dead tissue* and *dental plaque*. This may require a *local anesthetic*. In the first day after *treatment*, the *patient* may have to *rinse* the mouth with *hydrogen peroxide-based rinses* (3% *hydrogen peroxide* mixed with an equal amount of *water*) instead of *brushing* the *teeth*. [...]

Figure 3: A translated article about *acute necrotizing gingivitis* disease in the Dutch edition of the Merck Manual. The length of the first and third paragraphs have been reduced.

A PoS tag filter is a rule which selects a prespecified sequence of word classes. A basic example of such a rule is *Determiner Adjective Noun*. The use of PoS tag filters has been widely reported by previous studies [Bourigault, 1992; Daille et al., 1994; Justeson and Katz, 1995]. Some of these filters are very specific such as the one in Dagan and Church [1994] that allows only *Noun+* sequences. The precision of this kind of filters is high, but since the filters do not allow some prepositions that are frequently found in terms , such as *of*, their recall would tend to be low. However, other filters such as the ones used in Justeson and Katz [1995] allow more PoS tags and more possibilities of sequences. In our experiment, we use the filter by Justeson and Katz [1995] because we want to get a high recall, and let the next step (the statistical method, Section 2.4) improve the precision.

We slightly modify the Justeson and Katz filter for Dutch by adding an optional determiner to the original filter, because our terminology resources, such as the ICD-9 DE (the Dutch edition of NCHS [1996]), include terms with determiners (e.g., *degeneration of the choroids* and *stenosis of the larynx*, both of which satisfy the PoS tag sequence filter *Noun Preposition Determiner Noun*). We evaluate whether this addition will lead to better results or instead just introduce more noise.

All our texts have been analyzed by a syntactic parser which has identified syntactic structures such as noun phrases and verb phrases, as well as the relations between these phrases. Since we have this syntactic information at our disposal, we have also looked at the possibility of building syntactic filters. We have extracted word sequences marked as noun phrases from the corpus and selected these as candidate terms. For example, applied to the first sentence in the Figure 3, this filter produces the following candidate terms:

- *Acute necrotizing gingivitis Plaut-Vincent angina acute necrotizing ulcer-*

ative gingivitis

- *Acute necrotizing gingivitis*
- *Plaut-Vincent angina acute necrotizing ulcerative gingivitis*
- *acute necrotizing ulcerative gingivitis*
- *painful non-contagious infection of the gums*
- *gums*
- *pain fever*
- *pain*
- *fever*
- *fatigue*

The syntactic filter is more liberal than the PoS tag filter. It extracts candidate terms of any PoS tag sequence as long as the sequences are of category `noun phrase`. As shown in this example, the filter extracts a very long noun phrase (the first candidate, 10 words), which would not have been selected by the PoS tag filter. The reason for this is that the automatic syntactic analysis ignores punctuation signs such as commas and brackets while these are kept by the PoS filter. The syntactic filter is able to extract nested candidate terms, such as *Acute necrotizing gingivitis* and *gums*, which appear in longer noun phrases. Despite the removal of punctuation signs by the preprocessing step, the filter is able to identify the embedded noun phrase *acute necrotizing ulcerative gingivitis* and suggest it as a candidate term.

We tested both filters and found that the PoS tag filter performed better than the syntactic filter. The syntactic filter proposed more candidates (50,287–45,449) but the PoS tag filter achieved higher precision (37%–20%) and recall rates (62%–38%). Some examples of the proposed terms: *illness* (suggested by both filters), *kidney disorder* (only proposed by the Syntactic filter) and *obstruction of movement* (only found by the PoS tag filter).

2.4 Ranking Candidate Terms

In order to improve the precision of the term extraction process, we perform a second step after filtering candidate terms from text. Working on candidate terms consisting of two words, we rank candidates based on the frequencies of the individual words and the strength of the statistical association between them. We evaluate eight different methods for assessing association strength using uninterpolated average precision (see Section 2.1) for the first 500 terms as evaluation.

We calculated the association strengths using the output of the PoS filter. The baseline method for candidate term ranking is frequency, which assigns a higher rank to frequent word pairs. This approach achieved an uninterpolated average precision score of 67%. Of the seven other methods that we evaluated, the two best were chi-square (χ^2) and Dice's coefficient [Manning and Schütze, 1999, p.299]. Both achieved a precision score of 88% [Fahmi, 2008]. The same

three candidate terms appeared on the top of the two ranked lists: *magnetic resonance imaging*, *islets of Langerhans* and *sphincter of Oddi*, all of which are medical terms.

This approach works fine for terms consisting of two words. If candidate terms contain more than two words, for example *body mass index*, we may divide them into so-called pseudo bigrams. These are pairs of phrases which may contain one word or more than words. For example, *body mass index* can be divided in two pseudo-bigrams: *body – mass index* and *body mass – index*. The statistical methods can be applied to both pairs and for each method we choose the pair with the highest combined score.

The single-word candidate terms are a special challenge. The statistical methods based on association strength are obviously inapplicable. A popular alternative is to measure how specific the term is to a selection of texts. The intuition is that domain-specific terms will occur more frequently in domain-specific texts than in general texts. However, we refrained from using additional general texts.

The method which we developed for ranking candidate terms that consist of only one word, is based on the distribution of words in the ranked list of candidate multiword terms. We chose a rank threshold (in the range 2000–6000) and labeled all terms with a frequency above this threshold *relevant* while the rest were labeled *general*. Next, we designed a termhood score for single words which prefers words which frequently occur in relevant terms over words which frequently occur in general terms.

The candidate single-word terms were ranked according to the termhood scores and the ranked list was compared with a term list extracted from an annotated corpus. We compared five different rank threshold values and found that the algorithm performed best with a threshold value of 4000. We measured a precision of 56% at rank 9000, slightly better than the 54% achieved using the baseline method, which only used term frequencies. The top three of the ranked list of candidate terms associated with threshold value 4000 was *doctor*, *year* and *woman*. The first and the third were regarded as correct medical terms.

Similar term ranking procedures will be needed if one is dealing with historical texts.

2.5 Term Variation

Two different terms may refer to the same concept. This is called term variation. For example, *carcinoma* and *cancer* are variants and can be used interchangeably. Depending on the domain, terminological variants are estimated to account for 15-35% of the joint occurrence of the two (or more) terms involved [Daille, 2003]. It is a well-known phenomenon that needs special treatment. We wish to keep track of variants for several reasons, such as for indexing and retrieval [Jacquemin, 2001], conceptual structuring [Daille, 2003], and enhancing term extraction [Nenadić et al., 2004]. Besides occurrence in text, term variations also occur in controlled terminological resources, such as UMLS.

We use variation recognition mainly to detect synonyms since they frequently occur in our extracted relations. These synonyms are linked by coordinations and need to be recognized correctly. Consider, for example, the following sentence that contains a medical relation in our corpus:

> Leprosy is a contagious disease caused by the *leprosy bacillus* "*Mycobacterium leprae*" *or* "*Hansen bacillus*" named after the Norwegian physician Armauer Hansen who discovered it in 1873.

The sentence contains synonymic variants of the term *leprosy bacillus*, namely *Mycobacterium leprae* and *Hansen bacillus*. Medical text contains other frequent variation types, namely acronyms and abbreviations. Consider, for example, the following sentence from our corpus, which illustrates how acronyms also lead to term variation:

> *Tuberculosis*, abbreviated with *TBC*, or even *TB*, is formerly a very dreaded disease caused by the bacterium "Mycobacterium tuberculosis".

Fahmi [2008] discusses several types of term variations in different degrees of detail. In this section, we investigate one type of variation: synonymy. We describe our method for detecting synonymous words and evaluate the performance of the methods using a medical corpus.

Our method for finding synonyms is adapted from the DIPRE method [Brin, 1999]. We use text patterns to find pairs of synonyms. We incorporate syntactic information into patterns, as in Hearst [1992] and Pustejovsky et al. [2001]. The syntactic information is mainly for detecting terms that occur in the corpus.

The synonym extraction method consists of three consecutive processes within an iteration. The extraction starts with the injection of a small set of synonym pairs as seeds. The seed pairs can be selected manually from the corpus. Each pair is a tuple like `<term1, term2>` where *term1* and *term2* are the seed synonyms. Having a seed list, we apply the following steps, first to learn variation patterns, and next to extract synonym pairs from the corpus:

Step 1 The process searches for the occurrence of the seed tuples in the corpus and keeps the contexts surrounding the tuples. For example, from the phrase *tubercolosis (or TB)*, the pattern *X (or Y)* will be extracted.

Step 2 Next, the generated patterns are sought throughout the corpus to extract a new set of candidate synonym tuples that match the patterns.

Step 3 Then semantic compatibility scores are computed for the candidate synonym pairs based on their occurrence in predefined phrases on Web pages (see Fahmi [2008] for more details). The synonym pairs are ranked based on their compatibility scores and pairs with scores below a certain threshold are discarded.

Step 4 Finally, the extracted synonym pairs are used as a new seed list. The four processing steps are repeated until the number of the extracted synonym pairs satisfies an expected level of coverage or until no new pairs are found.

As in Hearst [1992] and Brin [1999], the size of the initial seed list can be very small, as few as 3 pairs can be sufficient. However, if the corpus is small, as ours indeed is, the frequency of the seeds in the corpus can be very low as well. They will generate a small number of synonym patterns, which subsequently

yield a small set of new synonym tuples. A possible solution to this problem is to increase the size of the initial seed list.

The extraction approach has been applied to a medical corpus, a collection of Dutch medical texts containing 57,004 sentences. We started with a seed list of three synonym pairs, *retina–netvlies* (English: *retina*), *septum–tussenschot* (*nasal septum*), and *poliep –vleesboom* (*polyp*). We compared two selection criteria: one based on unrestricted cooccurrence (P_{NEAR}) and one based on cooccurrence in prespecified text patterns (P_{or}). P_{NEAR} found more pairs (699–233) and achieved higher recall scores (85%–70%) but P_{or} obtained a higher precision score (61%–98%). Interestingly enough, the two methods made different errors, for example *lens–cataract*, suggested by P_{or} and *back–epidural*, put forward by P_{NEAR}. Fahmi [2008] contains more details.

2.6 Term Labeling

In the previous sections we have discussed approaches to term and variant recognition. The output of these processes are terms ranked by termhood scores. At this point, we do not know whether a term such as *tuberculosis* is a name of a disease, treatment, or virus. These labels are assigned by the next processing step, Term Labeling or Classification.

The goal of term classification is to disambiguate terms. Term classification may help to map a term to its position in an ontology or thesaurus, or to understand the roles of a term in a relation (e.g., as actors, sources, objects) [Hatzivassiloglou et al., 2001]. This section describes approaches to term classification and discuss their relevance to our tasks.

Term classification is often done with machine learning techniques. Many of these techniques are based on statistical models, such as Hidden Markov Models (HMMs) and naive Bayes. Other techniques, such as decision trees, rule induction, support vector machines (SVMs), and genetic algorithms [Manning and Schütze, 1999] can also be used for classifying terms.

Our term classification work is inspired by Nobata et al. [1999]. They compared two classification methods for classifying terms from MEDLINE abstracts, one that used a statistical model and another that applied a decision tree. The first method classified terms by computing the similarity of the terms to the distribution of words in a preclassified word list from databases. Since a word list is rarely complete, it can be extended with the output of a word clustering process. The second method used several feature sets including PoS tags, morphological information, and a list of words specific to the domain. They found that for the four classes that they were interested in, the statistical method was better in classifying terms from two classes while the decision tree performed better for the other two classes.

An interesting fact of the work by Nobata et al. [1999] was that they used a set of preclassified words from databases to classify terms from text. This is a useful approach for labeling medical terms since large medical term databases are available, like for example the UMLS Metathesaurus. It contains a large number of preclassified terms (2.10 million terms or 4.7 million term class labels). This decreases the problem of data sparseness, a problem involving lack of data which often restricts a successful application of machine learning techniques.

We evaluated the performance of the method in labeling the 2000 most frequent terms of the IMIX medical corpus with one of a set of eleven labels. The

Data set	training	testing
treats	4,251	15
has symptom	3,782	11
causes	3,136	21
has definition	3,496	11
diagnoses	1,103	25
occurs	942	28
prevents	429	10
All	56,654	350

Table 1: The relation types and the numbers of related sentences in the training and test set of the IMIX medical corpus. Sentences counted in the test set were positive examples among the 50 sentences randomly selected from the corpus.

system achieved an accuracy of 73.3%. Suprisingly enough, labeling multiword terms was easier (78.2%) than labeling single word terms (72.6%). The approach resulted in different accuracies for the eleven labels. The highest accuracy was 80.4% for the label *disease symptom* while the lowest was 34.8% for *disease feature*.

3 Relation Extraction

This section deals with the use of dependency information to extract relations from text. We pursue an approach in which we automatically learn text patterns from sentences that contain related terms. On the basis of these patterns, we extract new relation information from non-classified sentences. This method is evaluated on two different corpora, namely the IMIX medical corpus and a medical subset of the Wikipedia pages.

3.1 Resources

We use the IMIX medical corpus for learning relation patterns. The corpus consists of texts from a medical encyclopedia and a medical handbook and was described in Section 2.2. Human annotators identified and marked the terms and the relations occurring in the texts. Fahmi [2008] presents the seven different relation types that are present in the 57,004 labeled sentences. As shown in Table 1, the number of occurrences of the different relations types varies. The relation type treats is the most frequent while prevents is the least frequent.

For each relation, we randomly selected 50 sentences as test set. For the purpose of the evaluation, we reannotated the test set: only sentences that contain two fully specified arguments of the relation are considered to be relevant instances of the relation. Note that, since relation labeling was done at text level, information of a single relation may be spread over more than one sentence. Because our preprocessing method operates on sentence level, we are only interested in sentences which contain complete relations.

For testing the performance of the relation patterns, we used the text of lemmas in the category Health Care from Dutch Wikipedia (105,088 sentences). We parsed this corpus using the Alpino parser [van Noord, 2006], which results

	dependency structures			accuracies		
Relation type	**L1**	**L2**	**L3**	**L1**	**L2**	**L3**
`causes`	942	1,625	647	95%	90%	75%
`has definition`	4,102	6,118	657	95%	65%	30%
`occurs`	548	2,219	1,238	90%	80%	50%
`treats`	300	1,826	1,026	85%	60%	45%
`has symptom`	1,220	2,668	850	80%	30%	0%
`prevents`	24	171	470	75%	50%	50%
`diagnoses`	34	265	231	60%	60%	35%
All	7,170	14,892	5,119	91%	60%	41%

Table 2: Number of dependency structures retrieved from the WIKIPEDIA corpus and their estimated accuracies (from a random selection of 20) per relation with 2, 1 or 0 matching concept labels.

in dependency parse trees. As this material was not annotated, evaluation of precision was performed by manually checking the output of the relation extraction system.

We used a subset of 3,142,578 terms from the UMLS for classifying relation's arguments. 5% of the terms were in Dutch while the other 95% were in English.

3.2 Learning Relation Patterns

The first task in the relation extraction process is to find patterns that predict term relations in text. We learn these patterns from the annotated IMIX corpus in four steps:

1. First, we extract all annotated relation phrases from the texts, for example *RSI stands for repetitive strain injury*. Each phrase is divided in three parts: the two relation objects (*RSI* and *repetitive strain injury*) and the remainder of the phrase (*stands for*), which will be the candidate pattern.

2. Next, we use linguistic methods for identifying the main parts of the two relational objects (*RSI* and *injury*) and use the term labeling process (Section 2.6) for assigning labels to both (*disease* and *disease*).

3. Then, we collect the parses of each of the candidate patterns and compute a weight for each of them based on how often they occur within a relation phrase and how often they do not.

4. Finally, the weights are combined with the available labeling information to produce weighted relation patterns of the format (type, label_1, label_2, phrase, weight). An example of such a pattern is (definition, disease, disease, stand for, 0.80). This specifies that a phrase linking two diseases with *stand for* has an 80% chance to express the definition of the first disease.

3.3 Extracting Relations

We use the patterns (type, $label_1$, $label_2$, phrase, weight) for extracting new pairs of concepts that are related by the relation of the same type. In order to achieve this, we look in the corpus for dependency structures that contain the phrases specified in the pattern as well as terms that can be labeled with $label_1$ and $label_2$.

Ideally, the semantic types of both arguments of the retrieved dependency patterns match with both semantic types $label_1$ and $label_2$. However, while inspecting the data, we found some relevant triples that match neither or only one of the two semantic types. Therefore, we have relaxed the selection criterion and identified three different groups of relation supporting information:

Level 1: full match: the dependency structure matches the relation type and both labels

Level 2: single label match: the dependency structure matches the relation type and only one of the two labels

Level 3: relation-only match: the dependency structure matches only the relation type and none of the two labels

This approach serves two goals. First, we hope to increase the recall of the approach by relaxing the selection criterion. Second, we are interested in finding out how well each of the three schemes performs, especially in comparison with the other two schemes.

3.4 Evaluation

We evaluate our method using two resources described in section 3.1, namely the IMIX medical corpus (IMIX) containing 57,004 sentences, and a medical subset of Wikipedia (WIKIPEDIA) containing 105,088 sentences. The task is to learn and to extract relations for the medical relation types in Table 1.

We distinguish two phases in this process. In the learning phase, we derive the parameters (weights) of the extraction process from the annotated IMIX CORPUS. In the extraction phase, we derive new pairs of related concepts from the WIKIPEDIA corpus.

In the learning phase, we learn relation patterns from labeled sentences. We generate a relation model containing relation patterns and the corresponding scores. The IMIX corpus contains 3,136 training sentences. From them, we derive 3,803 relation patterns. An example of such a pattern is *disease is caused by disease*. The patterns do not occur frequently in the corpus. The example pattern is the most frequent one with 23 occurrences. There are only 23 patterns which occur six or more times.

For each of the seven relations `causes`, `has definition`, `occurs`, `treats`, `has symptom`, `prevents` and `diagnoses`, and for each of the three extraction levels, we randomly selected twenty dependency structures and manually evaluated their accuracy. The results can be found in Table 2. Extraction at level 1 (91% on average) proved to be much more accurate than at the other two levels (60% and 41%). However, most correct pairs were found at level 2 (60%*14,982 = 8,989), so including this selection criterion will boost the recall score.

An example of a correct pair proposed by level 1 is *sepsis is caused by an infection*. An incorrect pair found by level 3 is *fire is caused by a spark*. This example shows a frequent error in the output at level 3: the two concepts are related but they do not fit in the target medical domain.

4 Conclusions and Future Prospects

We have illustrated Computational Linguistics (CL) applications on text aimed at ontology extraction, examining both automatic term recognition and relation extraction. Both applications are useful for researchers that need to detect the content in large text reserves.

We have deliberately focused on information extraction (IE) in this snapshot of CL work on texts and how it might contribute to the history of science. We should add that there is other CL work which might also be interesting, e.g., work classifying the sorts of literature references used in scientific papers, confirming, contradicting, distancing, etc. [Lenhert et al., 1990]. As a further example of CL work with potential spinoffs for history and, in particular, History of Science, we mention the substantial body of work on automatic summarization of texts [Mihalcea and Ceylan, 2007, and references there]. There is no room to present this work in detail here, but, in addition to needing to identify content, summarization work is concerned with identifying those focused sections of texts in which novel contributions are identified succinctly.

But the work on information extraction is more likely to be interesting both because there is a more substantial community of researchers involved, and also because it provides a more direct reflection of the content of scientific papers. Zhang et al. [2007] uses CL techniques on a body of texts on American history to populate an ontology of historical events. It is a modest first step, but it suggests that the collaboration between CL and History is promising.

References

D. Bourigault. Surface grammatical analysis for the extraction of terminological noun phrases. In *Proceedings of the 14th conference on Computational linguistics*, pages 977–981, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

S. Brin. Extracting patterns and relations from the world wide web. In *WebDB '98: Selected papers from the International Workshop on The World Wide Web and Databases*, pages 172–183, London, UK, 1999. Springer-Verlag. ISBN 3-540-65890-4.

C. Cardie and J. Wilkerson. Guest editors' introduction. *Journal of Information Technology and Politics*, 5(1):1–6, 2008. Special Issue on "Text Annotation for Political Science Research".

I. Dagan and K. W. Church. Termight: Identifying and translating technical terminology. In *ANLP*, pages 34–40, 1994.

B. Daille. Conceptual structuring through term variation. In *In Workshop on Multiword Expressions. Analysis, Acquisition and Treatment. Proceedings of the ACL'03*, 2003.

B. Daille, E. Gaussier, and J.-M. Langé. Towards automatic extraction of monolingual and bilingual terminology. In *Proceedings of the 15th conference on Computational linguistics*, pages 515–521, Morristown, NJ, USA, 1994. Association for Computational Linguistics.

L. Dibattista. *Jean Martin Charcot e la lingua della neurologia*. Editore Cacucci, Bari, 2003.

I. Fahmi. *Automatic Term and Relation Extraction for Medical Question Answering System*. PhD thesis, University of Groningen, October 2008.

D. Giampiccolo, A. Peñas, C. Ayache, D. Cristea, P. Forner, V. Jijkoun, P. Osenova, P. Rocha, B. Sacaleanu, and R. Sutcliffe. Overview of the clef 2007 multilingual question answering track. In *Working Notes for the CLEF 2007 Workshop*. Budapest, Hungary, 2007.

V. Hatzivassiloglou, P. A. Dubou'e, and A. Rzhetsky. Disambiguating proteins, genes, and rna in text: a machine learning approach. In *ISMB (Supplement of Bioinformatics)*, pages 97–106, 2001.

M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proc. of the 14th COLING*, pages 539–545, Nantes, France, 1992.

G. Hirst and O. Feiguina. Bigrams of syntactic labels for authorship discrimination of short texts. *Literary and Linguistic Computing*, 22(4):405–419, 2007.

ISO. *ISO-704: Terminology work – Principles and methods*. International Organization for Standardization, 2000.

C. Jacquemin. *Spotting and Discovering Terms through Natural Language Processing*. MIT Press, 2001.

J. Justeson and S. Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1:9–27, 1995.

W. Lenhert, C. Cardie, and E. Riloff. Analysing research papers using citation sentences. *Cognitive Science*, 12:511–518, 1990.

C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, 1999.

R. Mihalcea and H. Ceylan. Explorations in automatic book summarization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 380–389, 2007. URL `http://www.aclweb.org/anthology/D/D07/D07-1040`.

NCHS. *Classifications of Diseases and Functioning & Disability*. National Center for Health Statistics, 1996.

G. Nenadić, S. Ananiadou, and J. McNaught. Enhancing automatic term recognition through recognition of variation. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

J. Nerbonne. *The Exact Analysis of Text*, pages A–xi–A–xx. CSLI, Stanford, 2007. Forward to 3rd ed. of F. Mosteller and D. Wallace *Inference and Disputed Authorship: The Federalist* [1]1964.

C. Nobata, N. Collier, and J.-i. Tsujii. Automatic term identification and classification in biology texts. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium*, pages 369–374, 1999.

J. Pustejovsky, J. Castao, B. Cochran, M. Kotecki, M. Morrell, and A. Rumshisky. Extraction and disambiguation of acronym-meaning pairs in medline. In *Medinfo*, 2001.

J. C. Sager. Term formation. In S. E. Wright and G. Budin, editors, *Handbook of Terminology Management*, volume 2, pages 25–41. John Benjamins Publishing Company, 1997.

G. van Noord. At last parsing is now operational. In P. Mertens, C. Fairon, A. Dister, and P. Watrin, editors, *TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles*, pages 20–42, 2006.

J. Zhang, I. Fahmi, H. Ellerman, and G. Bouma. Mapping metadata for semantic web for history (SWHi): Aligning schemas with library metadata for an historical ontology. In *International Workshop on Collaborative Knowledge Management for Web Information Systems*, pages 103–116, Berlin, 2007. Springer. LNCS 4832.