Keyword Extraction Rules Based on a Part-Of-Speech Hierarchy

Richard Khoury, Fakhri Karray, *Senior Member, IEEE*, Mohamed Kamel, *Fellow, IEEE*

***Abstract*— In this paper, we set out to present an original rule-learning algorithm for symbolic natural language processing (NLP), designed to learn the rules of extraction of keywords marked in its training sentences. What really sets our methodology apart from other recent developments in the field of NLP is the implementation of a hierarchy of parts-of-speech at the very core of the algorithm. This makes the rules dependent only on the sentence's structure rather than on context and domain-specific information. The theoretical development and the experimental results support the conclusion that this improved methodology can be used to obtain an in-depth analysis of the text without being limited to a single domain of application. Consequently, it has the advantage of outperforming both traditional statistical and symbolic NLP methodologies.**

I. Introduction

Natural language processing (NLP) approaches can be divided in two main classes, namely those based on statistical techniques and those based on symbolic techniques [8]. On the one hand, statistical NLP approaches, which rely on superficial statistical data such as word counts and co-occurrences, have the distinct advantage of being general enough to generate a coarse-level understanding of practically any text. However, because they ignore semantic information and forgoing more complex reasoning, these approaches are rather inadequate when it comes to extracting finer details. On the other hand, symbolic NLP tools can make use of extensive background knowledge and semantic information. This makes them capable of extracting precise information and relations contained deep inside text documents. Unfortunately, the same domain-specific knowledge that gives them their advantage also limits their applicability to the narrow field in which this knowledge is valid and relevant.

This study expands the current literature on NLP by developing a new methodology that combines the advantages of both statistical and symbolic NLP, namely that of being able to perform a fine-grain analysis comparable to that of symbolic NLP applications, while maintaining the generality that characterises statistical NLP algorithms. More specifically, we present in this work an original rule-learning algorithm for symbolic NLP, designed to learn the rules of extraction of keywords marked in its training sentences. Those keywords represent the information that a potential user or system designer seeks to extract from the sentences. As such, the information sought varies with each different application. To be sure, the algorithm developed in this study is designed to extract the main verb of the sentence, along with its subject and objects. But what really sets our methodology apart from other recent developments in the field of symbolic NLP is the implementation of a hierarchy of parts-of-speech at the very core of the learning algorithm. This new approach makes the rules dependent only on the sentence's structure rather then on its content as in ontology-based methodologies [3]. By the same token, it also makes the rules independent of context and domain-specific information.

To put in perspective the design and scope of our research, the paper is structured as follows. The next section offers a review of previous work in the field of statistical and symbolic NLP, and clarifies the study's relationship to the extant literature. Section III then presents a complete description of our methodology and makes the case for its original contribution. Experimental results generated through an implementation of this methodology are expounded and thoroughly analysed in Section IV, followed by some concluding remarks in Section V.

II. Literature Review and Background

The main differences between statistical and symbolic NLP methodologies can be effectively analyzed in terms of the width of their coverage and the depth of their understanding [8]. As we have explained in Section I, Statistical NLP approaches are, by their very nature, general enough to be applied to a wide array of domains with little modification, but restricted to extracting superficial data. It thus appears that they have a large width but a limited depth. On their part, symbolic NLP tools can perform a fine-grain analysis of a text, but cannot easily be ported to other documents of a different nature. In this sense, it can be said that these methodologies have a large depth but a limited width. This distinction between width and depth bears, in more cases than not, on the practical applications of these methodologies, as will become evident in the following paragraphs.

A popular application of statistical NLP is for the creation of automated question-answering (QA) applications. These tools are meant to look for the answer to a user's query in a large body of information, such as a database or the Internet. The three main approaches that underlie the creation of QA tools are presented in [12]. The first approach aims to find the exact answer to any question the user asks. In order to do

Manuscript received June 16, 2007.

Authors are with the Electrical and Computer Engineering Department, University of Waterloo, 200 University Avenue West, Waterloo, ON, Canada, N2L 3G1. (e-mail: khoury@pami.uwaterloo.ca, karray@pami.uwaterloo.ca, mkamel@uwaterloo.ca).

so, a QA system begins by parsing the user's question to find important keywords or text chunks, and then looks for occurrences of these keywords or chunks in the search material. An answer can be found, but only if it is explicitly present in the material and uses the same wording as the question. The second approach takes the alternative route of not providing an answer per say, but to return to the user a set of documents relevant to the query, in which the answer is likely to be found. Systems following this design philosophy use traditional statistical tools to compute the relevance of words in documents, and return the documents that are ranked highest given the keywords in the user's query. The third and final approach shares the goal of the first, to directly answer questions, but simplifies the problem by relying on question templates and on annotations in the documents the system searches through. As such, systems based on this approach can only answer questions for which they have a matching template and relevant annotations. Their main advantage comes from the fact that a small number of queries are often asked by users.

As for the tools that make use of symbolic NLP methodologies, they are favoured when there is a need to retrieve some very specific information from a corpus of specialised documents. A typical example of such tools is the multi-agent patent document analysis system described in [9]. This system has agents designed to extract information from patent documents, others designed to retrieve patents that a relevant to a user's query, and others still designed to analyse specific technical details of the patent in light of the user's specifications. These agents obviously require in-depth domain-specific technical information in order to realise their tasks. This information is provided by domain-specific thesauri and ontologies, which are constructed by other agents of the system with input from domain experts

The various examples presented above illustrate the advantages and shortcomings of statistical and symbolic NLP methodologies. The main advantage of tools engineered on the basis of statistical NLP is that they can be applied to documents from a wide array of domains with little or no modifications. Indeed, were it not for issues of scalability, the QA tools presented in [12] could even be successfully applied to the entire Internet at once. However, by construction, statistical tools are fundamentally confined to a very coarse level of information analysis, a fact that restrains their depth of understanding. To highlight this fact, recall that the approaches discussed in our previous examples are limited in their applicability. They are either restricted to question templates, they return whole documents likely to hold the answer, or they require that the answer be written out explicitly using the same wording as the question. On the other hand, tools designed in the context of symbolic NLP turn out to have a lot of substance. In particular, they have the distinct advantage of detecting and extracting very fine-grained information from documents, such as technical patent information [9]. However, the reliance of these symbolic tools on domain-specific information restricts the width of their coverage. In fact, the authors note in [9] that, without this domain-specific information, it would be absolutely impossible for their system to function properly.

The new methodology we propose in this paper seeks to integrate the width of coverage and depth of understanding characteristics of statistical and symbolic NLP methodologies. As we have mentioned in Section I, our methodology learns rules that are precise enough to extract specific information from individual sentences, hence matching the depth of other symbolic NLP methods. At the same time, these rules are based only on the sentences' structures, and are free from reliance on domain-specific information that is at the root of the limitation on width in symbolic NLP methodologies. Furthermore, since recent studies indicate that syntactic structure information can be ported from one domain to another [11], we can say that our methodology has a width comparable to that of statistical NLP approaches.

III. The Rule-Learning Algorithm

A. *The Learning System*

The rule-learning algorithm adopted in this research is a supervised learning system based on the work of Stephen Soderland in [3]. The fundamental idea behind Soderland's methodology is to devise the strictest rule possible to handle a sentence, and then relax it gradually in order to handle other, similar sentences. By applying the same principle, we have developed a new a methodology to learn the rules needed to extract the main verb of a sentence, along with its subject and objects.

Our learning system relies on a training corpus of sentences, in which the correct words to be extracted are identified. Since our methodology is centered on a part-of-speech hierarchy that will be explained in the next section, the first operation that the algorithm must naturally execute is the part-of-speech tagging of the sentence. This operation associates each word in the sentence with its correct part-of-speech tag from the Penn Treebank set [1]. To perform this tagging, we resort to an implementation of the Brill tagger [4], which is a simple yet efficient tagger that assigns to each word its most commonly used part-of-speech tag, and then uses nine transformation rules to correct common tagging mistakes. Indeed, Brill reports that this system achieves a tagging accuracy of 95.6% [5]. Once tagged, the training sentences are manually split into short segments containing at most one word to extract. Each of these segments thus becomes the strictest extraction rule possible for that word in that context. The set of all the aforementioned segments constitutes the training corpus of rules for the rule-learning system.

As with any step in any system which involves manually

handling data, the fore-mentioned splitting of the sentences into segments is a bottleneck in the implementation process. Solace can be found, however, in the fact that this is neither a hard nor knowledge-intensive task. Indeed, sentences present many "natural" split points, such as commas and conjunctions. Moreover, the split does not need to be very precise, and can be off by a word or two without negatively influencing the system, for reasons that will become apparent later on. It thus appears that the splitting process is a simple one; so much so, in fact, that it will be possible to automate it. Future work will focus on this task.

When the learning system receives a new sentence from the training corpus, it immediately proceeds to add the sentence's rules to its internal rule base, and then computes the similarity of all pairs of rules. The next step is for the most similar pair of rules to be merged together. This merger is accomplished by generalising one of the rules in order to incorporate the second. After the merger is completed, the keywords are extracted twice from the corpus of sentences, first by applying the pre-merger rule base, and secondly by using the post-merger rule base including the merged rule. Once that is done, the precision and recall ratios of both extraction processes are compared in order to determine the best rule base. If the first extraction turns out to have the higher ratios, the merged rule is rejected and the two rules are kept in the rule base without modification. By contrast, if the second extraction has the higher precision and recall ratios, the merged rule is retained in lieu of the original pair and its similarity to each of the other rules in the rule base is computed. Both outcomes, however, lead the algorithm back to the initial point on the loop, explained above, from where the learning process will continue running for as many iterations as needed.

In order to compute the precision and recall ratios, the system compares the keywords extracted by applying the rules to those marked in the correct sentence. If a keyword is correctly extracted by the rules, it counts as a true positive (*TP*). However, a keyword that is extracted by the rules yet does not appear in the correct sentence counts as a false positive (*FP*), whereas a keyword missed by the rules counts as a false negative (*FN*). Furthermore, if the system is trained to recognise different types of keywords, and encounters a keyword extracted as the wrong type, it counts it as both a *FP* and a *FN*. Once these three values are calculated, the precision and recall ratios are computed as per Equations (1) and (2) below.

$$precision = \frac{TP}{TP + FP} \tag{1}$$

$$recall = \frac{TP}{TP + FN} \tag{2}$$

One way to perform the comparisons and mergers of two rules mentioned previously consists in relaxing the constraints on each word in one of the rules until it includes the second rule, as suggested by Soderland [3]. This is done by raising the level of the class to which each word belongs a number of levels in an ontology of concepts. However, for the ontology to be valid, the rule-learning system must be confined to its specific domain, as in the case of Soderland's study. On the other hand, although our research is not intended to be limited to a single domain, it is still possible for us to define a general hierarchy of categories. To this end, we have designed a novel part-of-speech hierarchy, where the merger of two rules is done by raising the level to which each word belongs in this hierarchy towards more and more general grammatical categories. For example, the rules "the sites" and "the dogs" can be merged into "the (plural common noun)", and that rule can then be merged with "a house" to become "(determiner) (common noun)". Moreover, in this proposed hierarchy, the similarity between two rules simply becomes the average cost of raising each of their words to the levels in the hierarchy needed to merge the rules. This cost is function of the number of grammatical elements represented by the category, and of the semantic importance of the category. For example, nouns and verbs are more semantically important in a sentence, and therefore more expensive to merge, then adjectives, which are in turn more important and expensive than punctuation marks. A complete discussion of our part-of-speech hierarchy is presented in the next section.

To be sure, in the context of this research, the term "applying the rule base" refers to the process of extracting the subjects, objects and verbs from a corpus of sentences according to the rules. As part of this process, the algorithm we develop matches each part of the sentence to the most similar rule in the base. For instance, if a sentence contains the words "a duck", the algorithm would recognise that it must match it to the rule "(determiner) (common noun)" instead of, say, "to (infinitive verb)". This matching component of the extraction process will be described in greater detail in a later section. Once the matching is done, the subjects, objects and verbs that our system must find and that are already marked in the rules, can simply be extracted by identifying the matching words in the sentences. In the special case of a training corpus of sentences where the correct words to be extracted are known, the application of a rule base yields results that could be compared to the correct solution in order to estimate the precision and recall of the rule base.

B. The Part-of-speech Hierarchy

It is commonplace to define a *part-of-speech* as a linguistic category of lexical items (generally words) that share common syntactic or morphological characteristics. Grammarians most often divide the English language into eight parts-of-speech: the noun, verb, adjective, adverb, pronoun, preposition, conjunction, and interjection.

However, this traditional division, which can be traced back to the Ancient Greek grammars devised in the Second Century BC [2], is far from being unanimously accepted. Depending on what one wishes to consider an independent part-of-speech, other divisions have been proposed, some counting dozens of categories, if not well over a hundred. Some more extreme examples include the Lancaster-Oslo/Bergen Corpus and the London-Lund Corpus of Spoken English, that contain 135 and 197 part-of-speech respectively [6].

As mentioned earlier, this study's core contribution to the field of NLP is the design of an original part-of-speech hierarchy. Oddly, no such hierarchy has ever been developed before; this study seeks to fill that gap. Such a hierarchy would unify the many theories regarding the part-of-speech divisions. Indeed, the second-lowest level of the hierarchy is composed of dozens of narrow part-of-speech categories, but as we move to higher and higher levels in the hierarchy, these categories are gradually merged into an ever-smaller number of more general groupings.

Given the preceding description, it is logical to say that the lowest, most basic level of the hierarchy is comprised of the lexical items taken from the sentence. The second level of the hierarchy corresponds to the first stage of the generalization of the lexical items. At this level, lexical items are grouped in the 46 parts-of-speech of the Penn Treebank, discussed in detail in [1]. We have made one notable modification to the Penn Treebank, which is the addition of a special "blank" symbol that stands for a missing word. This modification will allow rules and segments of different lengths to be compared and matched together, as the extra words in the longer sentence will be paired with "blank" words in the shorter one.

At the second stage of generalisation, parts-of-speech that represent morphological variations of the same words, such as "singular common noun" and "plural common noun", are brought together into *morphological categories* that constitute the third level of the hierarchy. Moving two levels beyond this last one, we reach the level in the hierarchy where items are grouped into *lexical categories*. We have defined nine such categories for words, and three for the various symbols and non-native words that are commonly encountered in English texts. Each lexical category also includes the special "blank" symbol. The nine word categories we adopted closely follow the eight traditional parts-of-speech defined by grammarians. They are the noun, verb, adjective, adverb, pronoun, conjunction (which includes prepositions), interjection, determiner and auxiliary verb. The extra three non-word lexical categories are the punctuation, number and foreign word.

The level of the hierarchy that falls between the morphological and lexical categories defined above will be called the *sub-lexical category*. This level puts together items that belong to the same lexical category and which form a sub-group within this category. For example, the morphological categories "present participle verb" and "past participle verb" can be seen as the sub-lexical category "participle tense verb" within the "verb" lexical category.

The level of the hierarchy above that of the lexical category will be designated as the *super-lexical category*, because it is the one where the lexical categories are grouped into four super-categories. At this level, we distinguish between "core words", or words that carry their meaning independently of any other (noun, verb and foreign word), "modifier words", which are words that modify other words (adjective, adverb, interjection and auxiliary verb), "function words" with no real meaning, that serve a practical function within the sentence (determiner, conjunction and pronoun), and "non-words", symbols that are not words at all (number and punctuation). Finally, the most general level of the hierarchy is the *universe* level, where all items are grouped in a single category.

To help visualise all these concepts, a graphical representation of the complete part-of-speech hierarchy developed in this section is presented in Appendix A.

The reason why we stated, in the previous section, that the task of splitting sentences in segments does not need to be tremendously precise now becomes apparent. As similar rules are merged together in the rule base, the extra or missing words in some segments will be evened out in the merged rule, and replaced by more general categories of parts-of-speech. Important words that are sometimes missing in the segments will be replaced by lexical-level categories, which will allow the use of our blank part-of-speech to handle cases where the word is absent. On the other hand, extra words in a segment which are truly irrelevant to the rule will be generalized up to the Universe category, meaning that they can be anything at all.

C. Applying the Rules

The next major building block in our methodology is the procedure to apply the rules to a corpus of sentences, in order to extract the desired keywords. This particular procedure serves a dual purpose. In the first place it serves to compute the precision and recall values of the various rule bases generated by the learning algorithm described previously, in order to compare them and pick the best one. Secondly, it serves to analyze real sentences using the final rule base learned by the algorithm.

It should be noted that, in the context of this study, the rules that make up the rule base are only a few words long, which makes them much shorter than normal sentences. In the circumstances, it can easily be seen that it will take a combination of several rules, in most cases, to cover an entire sentence. Obviously, a large number of such combinations is possible, and the best combination of rules to apply is the one that is most similar to a sentence, using the measure of similarity defined before. In this way, the problem of applying the rules becomes one of finding the most similar sequence of rules to cover a sentence.

This problem can be likened to that of searching a tree, in which each node is the choice of a rule, the cost of that node is the similarity of that rule to the part of the sentence to which it is being compared, and the cost of getting to that node is the similarity of the sequence of rules up to that point. The tree is searched using a uniform-cost search algorithm. This algorithm finds the cheapest node (i.e. finds the sequence of most similar rules up to a point in the sentence), expands it (i.e. applies all rules at that point), and repeats this process until the cheapest node is at the end of the sentence. The advantages of the uniform-cost search are that it is optimal and complete [7], which means that it is guaranteed to find the cheapest sequence of rules for each sentence.

IV. Experimental Results

A. Setup

As noted earlier, the algorithm developed in this study can be trained to extract any desired information from a text, provided the information could be obtained in the form of keywords in the sentences. To demonstrate this feature, the system was trained so as to extract the main verb of a sentence, along with the nouns that are its subjects and objects. However, the fact that most sentences typically contain many verbs and nouns that must be ignored constitutes a serious impediment to this extraction process. For example, in the sentence "John is prepared to leave for Berlin", the verb to leave is the main verb that the system must extract, while the other two verbs should be ignored. Similarly, in the sentence "The president of the country made a speech", the correct subject to extract is president, not country. Hence, the distinction between the nouns and verbs that should be retained and those that should be discarded is one of the main challenges that the extraction process must be capable of handling. The following section will demonstrate how our algorithm can effectively address this challenge.

The data used to train our system comes from the Brown Corpus [10]. This dataset is a corpus of American English written texts compiled in 1961. It is composed of 500 sample documents selected to reflect the spread of domains that were of interest to the American public at that time. The documents in the corpus thus cover a wide range of topics, from news coverage to religious texts, from industrial reports to detective fiction. By adopting this entire corpus as training data, our system should be able to learn rules to handle a wide variety of sentences reflecting many different writing styles. From a practical perspective, we decided to set off the demonstration by initially limiting the scope of the rule-learning system to the business domain of the corpus (samples A26-A28), and to train our system with a random sub-sample constituted from 10% of that domain. This sub-sample represents 269 training rules, or 27 sentences.

B. Results and Discussion

It is interesting to examine in the first place the behaviour of the number of rules in the rule base even as the learning process evolves. Figure 1 illustrates the size of the rule base after each training sentence has been processed. As the figure shows, the rule base quickly grows to the 20-rules range, but its rate of growth slows down considerably after this. To be sure, the rule base still gains new rules after reaching the 20-rule level, but it does so at a slow and irregular pace, and even eliminates rules at some points in the training process. In short, after learning 20 rules from the first 50 training rules, the learning process only discovers 13 additional ones from the next 220 training rules. This result seems to indicate that there is a limited number of rules to be learned, or alternatively that there is a limited number of atomic sentence structures that are combined in various ways as needed to form complete sentences.

To gain a deeper understanding of the learning process, it would be informative to analyze the behaviour of the precision and recall values of the rule base after each training sentence has been processed. To compute these values, we applied the rule base on the unseen 90% portion of the Brown Corpus' business domain that was not used in the training process. Overall, the results, which are illustrated in Figure 2, show that both the precision and recall values increase as the system becomes more and more trained. More specifically, the recall value starts off higher than the precision value but falls during the processing of the first 80 training rules, even as that of the precision increases. Figure 2 also reveals a sharp temporary rise of a few percentage points in the recall value in the 150-220 training rules range, which is matched by a corresponding drop in the precision value. This is a typical trade-off in classification systems, where an increase in precision is usually offset by a decrease in recall, or vice-versa [3] [13]. It is however interesting to note in this regard that, on the whole, the recall value shows a slight upward trend throughout the learning process. The precision value itself has a slow and irregular rate of increase, but gains nonetheless 10% from the start to the finish of the training process. These results indicate that, with further training, both the precision and recall values of our system could be improved.

The final results of the test of our algorithm using the unseen 90% portion of the Brown Corpus' business news

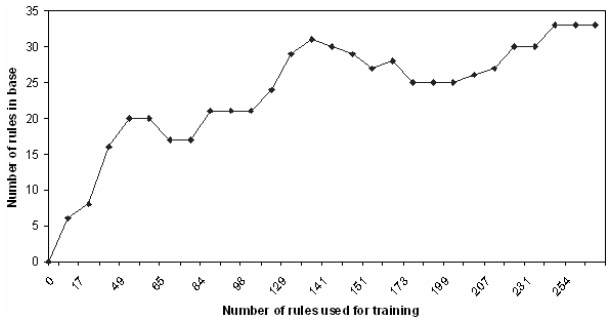


Fig. 1. Size of the rule base during training.

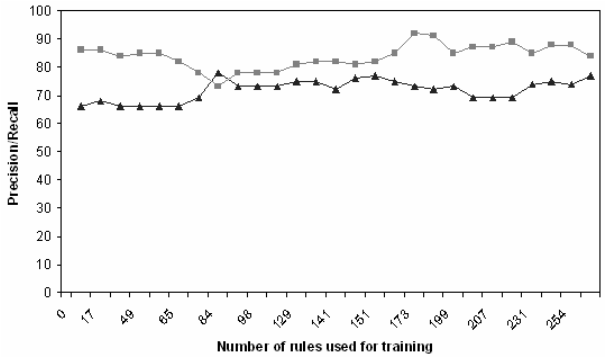


Fig. 2. Precision (triangles) and recall (squares) of the rule base during training.

domain are provided in Table 1. In order to fully illustrate the scope and usefulness of our methodology, the final rule base was subjected to two more tests beyond that one. To conduct these two extra tests, transcripts of television programs were selected. The first transcript comes from a normal, scripted television program[1], while the second comes from an unscripted debate show[2]. The reason for this selection of test corpora is that they are quite different from the business domain that was used to initially train the system. By design, the purpose of these tests is to demonstrate that the rules learned with our methodology are not domain-dependent. The results of the tests are provided in Table 1. For comparison purposes, the table also includes the precision and recall values of both a statistical and a symbolic NLP algorithm. The symbolic algorithm represented here is the one developed by Soderland [3] which, as we note, underlies our work. The experiments done in [3], the results of which are include in Table 1, were performed on a corpus of business news articles similar in nature to the portion of the Brown Corpus we selected, and thus their results are consistent with our study. The statistical algorithm selected is the Minipar parser, which was chosen because was trained using the Brown Corpus to extract the same kind of information for which our algorithm has been trained, and also because it has been thoroughly studied and evaluated in [14].

As can be observed in Table 1, our methodology compares quite favourably with the two reference algorithms. In all three tests, our system exhibits a better recall than the two references, and in one test a better precision than Soderland's algorithm. Moreover, our algorithm's field of application seems to be neither limited in width nor in depth as is the case for the two algorithms in reference. Indeed, while Minipar is quite good at the task of finding the subjects and objects of sentences, its

[1] We selected for that purpose the pilot episode of the American sitcom "That '70 Show". The transcript is available online at http://www.twiztv.com/scripts/that70sshow/season1/that70sshow-101.htm.

[2] The debate chosen is the one that took place between Jon Stewart, Tucker Carlson and Paul Begala on the CNN program "Crossfire". The transcript is available online on CNN's website, at http://transcripts.cnn.com/TRANSCRIPTS/0410/15/cf.01.html.

TABLE I
EXPERIMENTAL RESULTS OF OUR ALGORITHM (ROWS 1, 2 & 3) AND TWO REFERENCE ALGORITHMS (ROWS 4 & 5).

	Precision	Recall	Limited width?	Limited depth?
Business domain	77%	83%	No	No
TV show	62%	91%		
Debate show	61%	90%		
Statistical algorithm	88%	75%	No	Yes
Symbolic algorithm	70%	56%	Yes	No

performance degrades severely when it is used to extract information that requires a deeper understanding of the text. For example, on the task of finding conjunctions, Minipar only achieves a precision and recall of 67% and 50%, while for relative clauses those values drop even more to 52% and 56%. This rather poor performance is due to the fact that Minipar relies on statistical information, which by its very nature only gives an overall view of the text and is not always well-suited for the extraction of specific details from sentences (as explained in Sections I and II). This limitation of Minipar contrasts with the capacity of our algorithm to successfully extract more specific information from the sentences, by simply adding new and more specialized rules to its rule base. On the other hand, the width limitation of symbolic algorithms, such as Soderland's, comes from their reliance on domain-specific ontologies or other specialized sources of information. This type of information cannot be ported to other domains, which severely limits the scope of algorithms that are based on it. By contrast, our algorithm does not rely on any such constrained information, and the positive results we obtain when we apply the rule base extracted from the business domain to science-fiction domain sentences clearly demonstrate that our methodology is not domain-specific. Taken together, these results suggest that the algorithm developed in this study not only performs as well as statistical and symbolic NLP algorithms, but also exhibits a wider and deeper field of application then either one of these other algorithms.

Interestingly, the precision and recall results are almost exactly the same for the scripted TV show dialogue and for the improvised TV debate, and in both cases they are rather different from those obtained with the business documents, with the recall value being higher and the precision value lower for the TV transcripts. Our analysis provides insight into the reasons behind these observations. Notably, the higher recall value is contrary to the expectation that the rule base should perform better on unseen in-domain data than on out-of-domain data. The explanation for this surprising result is that the sentences found in both transcripts are typically simpler than those in business-related news articles. Indeed, spoken sentences, regardless of whether they are scripted or improvised, are typically shorter and more straightforward than written sentences. This allows the rules to recognise the keywords more reliably and to miss

fewer of them, thus improving the recall value. Unfortunately, spoken sentences also make use of sentence structures that are not found in business news articles, and for which the rule base has no corresponding rule. For example, spoken sentences can make heavy use of interjections, and the speaker often repeats words for emphasis. Without the correct rule to handle such sentences, the system will try to apply the most similar rule found in the rule base. This most similar rule, however, may not be appropriate, and may mark the wrong word as keyword. This causes the drop in precision value.

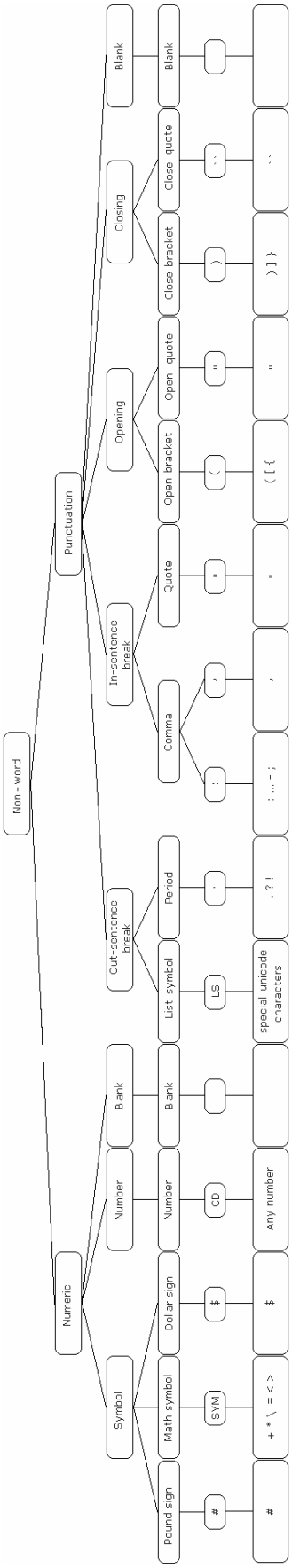
For the sake of completeness, it is worth noting the main drawback of our methodology at this point, which is its requirement in computational time. Indeed, in the upper-bound case, each word of a sentence segment will be compared to each word of each rule in the rule base in order to find the most similar rule to that segment, which leads to a cubic algorithmic complexity. Clearly, optimising the search algorithm will be an important focus in future research.

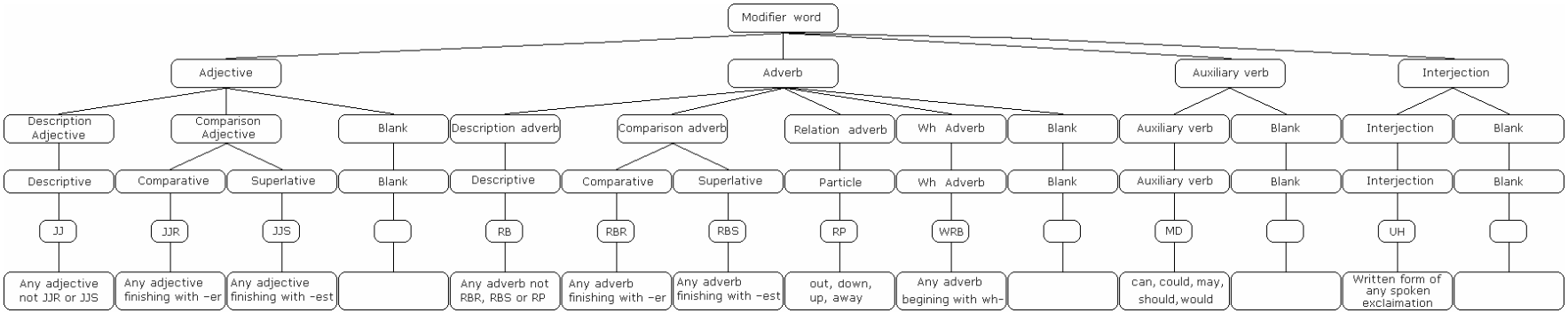
V. Conclusion

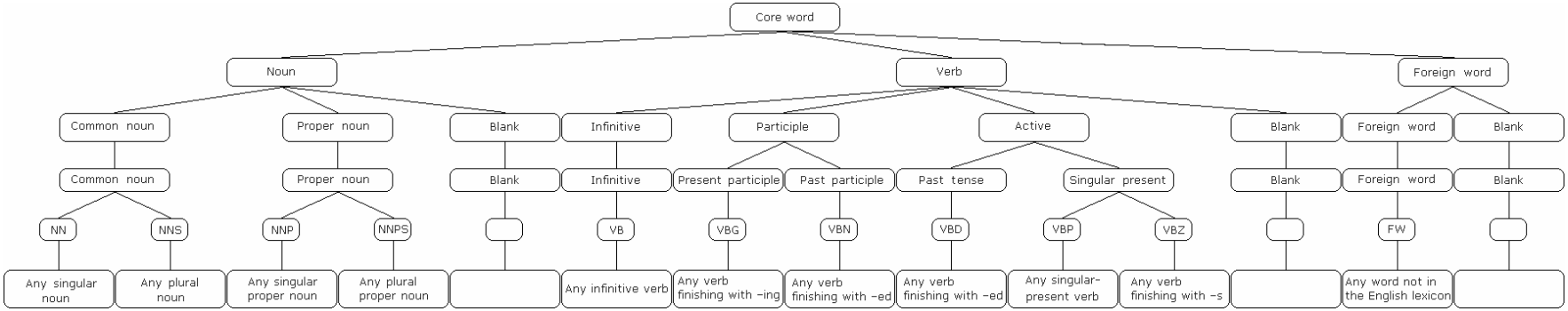
This paper has introduced the concept of a part-of-speech hierarchy, and applied it in the context of a symbolic NLP methodology designed to extract information-rich keywords from English sentences. This methodology learns the set of rules to extract the desired keywords based on a corpus of annotated training examples. Following this learning stage, the methodology is able to extract the keywords from any sentence, by matching it to the most similar sequence of rules. The part-of-speech hierarchy we proposed is essential in this methodology. By providing a metric to compute the distance between words, the hierarchy enables the system to compare rules, compute the similarity measure, and perform the learning. In addition to the theoretical development of our methodology, experimental results are provided and thoroughly analysed in order to confirm that our methodology can be used to obtain an in-depth analysis of texts without being limited to a single domain of application. Putting these arguments together leads to the conclusion that the proposed methodology should outperform both traditional statistical and symbolic NLP methodologies.

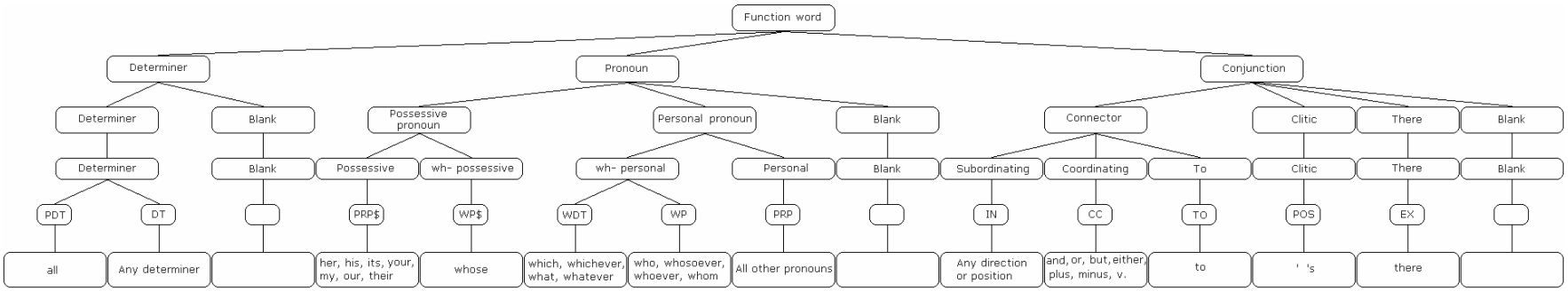
VI. Appendix A

Due to space restriction, we show here each of the four super-lexical categories separately, along with all their sub-levels. The top level, in which these four super-lexical categories are grouped together in the single “Universe” category, is not shown here.




Modifier word
Adjective
Adverb
Auxiliary verb
Interjection
Description Adjective
Comparison Adjective
Blank
Description adverb
Comparison adverb
Relation adverb
Wh Adverb
Blank
Auxiliary verb
Blank
Interjection
Blank
Descriptive
Comparative
Superlative
Blank
Descriptive
Comparative
Superlative
Particle
Wh Adverb
Blank
Auxiliary verb
Blank
Interjection
Blank
JJ
JJR
JJS
RB
RBR
RBS
RP
WRB
MD
UH
Any adjective not JJR or JJS
Any adjective finishing with -er
Any adjective finishing with -est
Any adverb not RBR, RBS or RP
Any adverb finishing with -er
Any adverb finishing with -est
out, down, up, away
Any adverb begining with wh-
can, could, may, should, would
Written form of any spoken exclaimation


Core word
Noun
Verb
Foreign word
Common noun
Proper noun
Blank
Infinitive
Participle
Active
Blank
Foreign word
Blank
Common noun
Proper noun
Blank
Infinitive
Present participle
Past participle
Past tense
Singular present
Blank
Foreign word
Blank
NN
NNS
NNP
NNPS
VB
VBG
VBN
VBD
VBP
VBZ
FW
Any singular noun
Any plural noun
Any singular proper noun
Any plural proper noun
Any infinitive verb
Any verb finishing with -ing
Any verb finishing with -ed
Any verb finishing with -ed
Any singular-present verb
Any verb finishing with -s
Any word not in the English lexicon


Function word
Determiner
Pronoun
Conjunction
Determiner
Blank
Possessive pronoun
Personal pronoun
Blank
Connector
Clitic
There
Blank
Determiner
Blank
Possessive
wh- possessive
wh- personal
Personal
Blank
Subordinating
Coordinating
To
Clitic
There
Blank
PDT
DT
PRP$
WP$
WDT
WP
PRP
IN
CC
TO
POS
EX
all
Any determiner
her, his, its, your, my, our, their
whose
which, whichever, what, whatever
who, whosoever, whoever, whom
All other pronouns
Any direction or position
and, or, but, either, plus, minus, v.
to
' 's
there

VII. Acknowledgment

We would like to thank Professor Chrysanne DiMarco for her helpful comments and valuable insights.

References

[1] B. Santorini, "Part-of-speech tagging guidelines for the Penn Treebank Project", Technical report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania.

[2] I. Michael, "English grammatical categories and the tradition to 1800", Cambridge University Press, 1970.

[3] S. G. Soderland, "Learning text analysis rules for domain-specific natural language processing", Tech. Rep. UM-CS-1996- 087, 1996.

[4] E. Brill, "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging", in *Computational Linguistics*, vol. 21, 1995, pp. 543-565.

[5] E. Brill, "Unsupervised learning of disambiguation rules for part of speech tagging", in *Proceedings of the Third Workshop on Very Large Corpora (WVLC 3)*, 30 June 1995, pp. 1-13.

[6] M. Marcus, B. Santorini and M. A. Marcinkiewicz, "Building a large annotated corpus of English: the Penn Treebank", in Computational Linguistics, vol. 19, no. 2, 1993, pp. 313-330.

[7] S. Russel, P. Norvig, "Artificial intelligence: a modern approach", Prentice Hall Series in Artificial Intelligence. Englewood Cliffs, New Jersey, 1995.

[8] M. Shamsfard and A. A. Barforoush, "The state of the art in ontology learning: a framework for comparison", in *The Knowledge Engineering Review*, vol. 18 no. 4, 2003, pp. 293-316.

[9] V.-W. Soo, S.-Y. Lin, S.-Y. Yang, S.-N. Lin, and S.-L. Chen, "A Cooperative multi-agent platform for invention based on ontology and patent document analysis", in *Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design*, 24-26 May 2005, vol. 1, pp. 411-416.

[10] W. N. Francis and H. Kučera, "Manual of information to accompany a standard corpus of present-day edited American English, for use with digital computers", Department of Linguistincs, Brown University, Providence, Rhode Island, 1964.

[11] C. Chelba, "Portability of syntactic structure for language modeling", in *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, 7-11 May 2001, vol. 1, pp. a-d.

[12] A. Andrenucci, and E. Sneiders, "Automated question answering: review of the main approaches", in the *Third International Conference on Information Technology and Applications*, 4-7 July 2005, vol. 1, pp. 514-519.

[13] F. Sebastiani, "Machine learning in automated text categorization", in *ACM Computing Surveys*, vol. 34, no. 1, 2002, pp. 1–47.

[14] D. Lin, "Dependency-based Evaluation of MINIPAR", *in Proceedings of The Evaluation of Parsing Systems: Workshop at the 1st International Conference on Language Resources and Evaluation (LREC'98)*, Granada, Spain, 28-30 May 1998.