Improvement of Named Entity Tagging by Machine Learning

Thamar Solorio

Reporte Técnico No. CCC-04-004
March 12th, 2004

Luis Enrique Erro 1
Sta. Ma. Tonantzintla,
72840, Puebla, México.

Improvement of Named Entity Tagging by Machine Learning

Thamar Solorio[1]

[1]Coordinación de Ciencias Computacionales,
Instituto Nacional de Astrofísica, Óptica y Electrónica,
Luis Enrique Erro 1, Sta. Ma. Tonantzintla,
72840, Puebla, México
thamy@inaoep.mx

Abstract
A Named Entity (NE) is a word, or sequence of words that can be classified as a name of a person, organization, location, date, time, percentage or quantity. Named entities can be valuable in several natural language applications. For instance, automatic text summarization systems can be enriched by using NEs, as they provide important cues for identifying relevant segments in text. Other uses of NE taggers are in the fields of information retrieval (i.e. more accurate Internet search engines), information extraction, automatic speech recognition, question answering and machine translation.

There has been a considerable amount of work that aims to improve the performance of NE taggers, however most of these efforts are targeted to build handcrafted NE taggers. While handcrafted systems can achieve good performance they have several disadvantages, such as the need of rebuilding the NE tagger in order to port it to new domains. This proposal presents a research project for building an automated method, based on machine learning, that facilitates the portability of handcrafted NE taggers. The hypothesis underlying this proposal is that the coverage of NE taggers can be increased using machine learning techniques, without the need of rebuilding the linguistic resources of the NE tagger. One motivation for this research is that the task of building a training set is faster and easier than building the linguistic resources for the handcrafted systems. Preliminary results presented here show that it is feasible to prove the correctness of the hypothesis.

1. Introduction

1.1 Overview and Motivation

Information Extraction is the task of extracting specific kinds of information from documents [1]. As an example, suppose that we have a collection of disaster-related documents and we are interested in analyzing them with several purposes. We might be interested in extracting from the documents some very specific parts such as where and when the disaster occurred, how many casualties and material losses. While some of these data, such as date and numeric expressions, can be easily detected by automated means, other, such as locations, are more complicated and might need the use of natural language processing techniques. A Named Entity (NE) is a word, or sequence of words that can be classified as a name of a person, organization, location, date, time or quantity. In Information Extraction systems, accurate detection and classification of NEs is a very important task given that NEs can help us to extract knowledge from texts; such as where the event happened, who were involved and when it happened.

Named Entities can be valuable in several Natural Language applications. For instance, extracting NEs can be a first step toward automating the laborious task of ontology building. Mann uses a proper noun ontology in order to improve the performance of a Question Answering system [2]. Automatic Text Summarization Systems can be enriched by using NEs, as they provide important cues for identifying relevant segments in text [3]. Other uses of NE taggers are in machine translation systems where proper nouns can be translated with higher accuracy using NE taggers [4]. In the field of information retrieval NEs can help to build more accurate internet search engines.

While for a human the task of identifying in a text which parts correspond to NEs can be trivial, the same cannot be said about designing programs capable of classifying named entities with human-level performance. This is a difficult goal to achieve due to a common problem of all natural language processing tasks: ambiguity. Another inconvenience is that documents are not uniform, their writing style, as well as the vocabulary can change dramatically from one document to another. Thus, there has been a considerable amount of work that aims to improve the performance of NE taggers.

However, most of the previous methods require a lot of human effort, as they heavily rely on handcrafted linguistic resources, such as context free grammars, regular expressions, lists of trigger words, gazetteers or dictionaries [5-9]. Even though the accuracy of these handcrafted systems can be high we can discuss several disadvantages:

- The development of the linguistic resources requires a considerable amount of time, a lot of human effort, and a significant computational linguistic knowledge.
- The complete effort of having an NE tagger running is wasted when the NE tagger needs to be ported to a new domain or language. As mentioned earlier, features of documents in one domain are different from those of documents belonging to a different domain. It is very likely that the handcrafted rules that perform well on a given collection will not have the same performance on a different one.
- Another disadvantage related to the portability of the NE tagger is that the regular user of the handcrafted system may not have the linguistic knowledge needed in order to adapt the system, so the user will need a large amount of time in order to learn every detail of the system before she can start the process of adaptability.
- Even though there can be a lot of time invested on carefully designing the linguistic resources, there will always be exceptions that can cause these resources to fail. It is generally impossible, given the usual time constraints, even to code for every exception which one can think of, leaving aside those exceptions which do not become apparent until one has run a test [1].
- Finally, it must be remarked the clear dependency of the handcrafted system performance and their human designers. Any bias the human designer may have when constructing the system will be inevitably acquired by its design. This is in most cases undesirable.

It is clear that new methods that overcome the limitations of the handcrafted systems need to be developed. These new methods require the ability of training themselves in order to eliminate the need of expensive human effort. Thus, it is not surprising that researchers that once used handcrafted systems are turning to machine learning algorithms that can be trained automatically to perform NE classification [1,10,11].

This document describes proposed research to develop a method for Named Entity classification using machine learning techniques. The method will be capable of automatically recognize and classify NEs in unrestricted text. The overall goal of this research is to develop an automated method that facilitates the portability of NE taggers to new domains and/or languages without the effort needed by the handcrafted systems. Various experiments have been performed and preliminary results show that it is possible to build accurate NE taggers using machine learning algorithms.

This document is organized as follows. Next section summarizes related work. Section 2 presents the preliminary research work, Section 3 introduces the research question of this thesis proposal. Section 4 describes the methodology as well as preliminary research work, and finally Section 5 discusses possible contributions of the research.

2. Related Work

This section surveys previous work related to the problem of NE recognition and classification in unrestricted text together with work related to relation extraction. The latter was considered due to the concern of working in this direction if time allows. The review of the state of the art described here is by no means exhaustive, as this is an ongoing research area. This section covers only the more closely related previous work.

2.1 Named Entity Recognition and Classification

In [12] a new method for automating the task of extending a proper noun dictionary is presented. The method combines two learning approaches: an inductive decision-tree classifier and unsupervised probabilistic learning of syntactic and semantic context. The decision tree approach learns, from an initial proper noun dictionary and a training corpus, to assign semantic categories to proper nouns that are not in the dictionary. In this stage only high confidence classifications were accepted. On a second stage, unsupervised learning is used to improve recall. The attribute information selected for the experiments uses Part of Speech (PoS) tags as well as morphological information whenever available. Also, they do not use the original proper noun words, at least not completely. They use the first two and last two words of the proper noun, along with contextual information (two words to the left of the proper noun and two words to the right).

Sekine et al. [13] presented a named entity hierarchy containing 150 NE types. The purpose is to make available this resource in order to develop applications of information extraction, question answering and also to further extend this hierarchy in some specific domains. In order to construct the hierarchy they considered the following issues:

- They used the surface form as the primary clue in ambiguous cases.
- They considered class names like "color" as part of the hierarchy while leaving out ordinary words like "cup".
- The degree of fineness depends primarily on the context rather than the name itself.

The procedure for building the hierarchy consists of three steps:

1. The use of three methods for building three initial hierarchies: corpus-based, based on previous systems and tasks, and based on thesauri.
2. Merge the three hierarchies.
3. Refine the hierarchy by tagging additional corpus and developing automatic taggers.

A very interesting system for NE classification based on Hidden Markov Models was proposed by Zhou and Su [7]. The novelty in this system is the combination of information used to build the HMM-based tagger: they use a combination of internal and external sub-features. The internal sub-features are three: in the first one they consider information relevant to discriminate between dates, percentages, times and monetary amounts, also capitalization information; the second sub-feature is about important triggers that the authors regard useful for NE recognition; the last internal sub-feature contains gazetteer information, gathered from look-up gazetteers that contain lists of names of persons, organizations, locations and other kind of named entities. The external evidence refers to context of other NEs already recognized. A list is updated with every named entity that has been recognized so far. When a new candidate is found an alias algorithm is invoked to determine its relation with the NE in the list.

One work focused in NE recognition and classification for Spanish is based on discriminating among different kinds of named entities (NEs): core NEs, which contain a trigger word as nucleus, syntactically simple weak NEs, formed by single noun phrases, and syntactically complex named entities, formed by complex noun phases. Arévalo et al. [6] focused on the first two kinds of NEs. The method is a sequence of processes that uses simple attributes combined with external information provided by gazetteers and lists of trigger words. A context free grammar, manually coded, is used for recognizing syntactic patterns.

Carreras et al. presented a method that uses an ensemble of decision trees in order to learn NE recognition and classification [14,10]. The recognition phase is performed combining three schemes: a BIO classifier, where each word is tagged with one of three tags, beginning of NE (B), inside the NE (I) or outside (O); an open-close&inside method that detects the word that opens and the word that closes the NE; the third method is global open-close, where the predictions of an open and a close classifier are combined to achieve a final prediction. The classification phase was performed combining with Error Correcting Output Coding [15] binary classifiers. Among the external information used for learning NE classifiers are: lists of trigger words and a gazetteer.

In [16] the authors experimented with a risk minimization approach. They performed several experiments, combining different features. Their best performance for NE recognition in English is achieved by a system that combines dictionaries and trigger word lists with linguistic features extracted directly from the text, such as case information, token prefix and suffix. However, in the German data set the improvement from using additional information was not that great. The authors believe that this difference between the English and German data sets may be due to the fact that, for English, there is a higher availability of good quality dictionaries.

2.2 Discovering Relations

Natural language processing systems require a variety of different kinds of knowledge, i.e. lexical inventory, morphological and syntactic rules or constraints, semantic and conceptual knowledge. But while all these knowledge types are provided by humans, there are few efforts for building systems that allow simultaneous learning of different types of relevant linguistic knowledge. In [17] they propose an integrated approach for learning syntactic and conceptual knowledge simultaneously. The system they proposed allows the automated maintenance and growth of domain-specific concept taxonomies and grammatical class hierarchies simultaneously, based on knowledge captured from natural language texts.

Basu et al. defined a metric for evaluating the novelty of text-mined rules that considers semantic information gathered from the lexical knowledge-base of English WordNet [18]. So the novelty in a rule is proportional to the distance between the terms in the hierarchy of WordNet. The presented experimental results showed that this approach for scoring text-mined rules correlates almost in the same manner that human judgements correlate with each other.

Rosario and Hearst [19] report a method based on a neural network that classifies semantic relations of two terms noun compounds. Their method uses a domain specific hierarchy and they experimented on different levels of generality. They report an accuracy of 60%, with 18 classes of semantic relations where only 75 examples out of 805 test instances contained noun compounds in which both words where not present in the training set. A more realistic experiment, where only around 6% of the testing noun compounds where present in the training set, showed an accuracy of 46%.

In [20] Girju et al. present a method that uses decision tree learner C4.5 to obtain semantic constraints for discovering Part-Whole relations. They make use of WordNet in order to build a training set containing positive and negative examples of part-whole relations. These examples are the inputs for C4.5 learning algorithm, and the outputs are the semantic constraints. Basically they pose the problem as a binary classification learning problem where a training set is used to learn a decision tree classifier which is then used to classify unseen examples. Their system achieves a precision of 83%.

2.3 Named Entities and Relations in Text

A probabilistic reasoning for recognition of entity and relation was proposed by Roth and Yih [21]. As opposed to common approaches where the problems of recognizing entities and relations are treated separately, they developed a method for identifying entities and relations together. In this framework they train weak classifiers for recognition of entities and relations independently based on features such as PoS-tags, words, conjunctions of words, bigrams and trigrams among others. These weak classifiers output a probability estimation which is in turn used to feed a belief network. The belief network, that represents the constraint inferred from the training set, is used to compute the most probable global predictions of the class labels. These global predictions are computed using the belief propagation algorithm. Experimental results show that the proposed approach presents a good alternative for solving the problem of identifying entities and relations with an acceptable accuracy. However, they performed tests with only two kinds of relations, and the entities involved were also two. It is very likely that adding a more realistic number of relations and entities might degrade the performance of this method.

3. Research Question

As mentioned earlier, accurate classification of NEs in unrestricted text is an important first step toward building robust natural language processing systems. Considering that NEs are an open class, it is desirable to have automated means for adapting the possible low coverage of the handcrafted systems. This paper presents a research project which aims to develop an automated method for decreasing the effort of porting a handcrafted NE tagger to a new domain. The hypothesis that this research will try to prove is the following: *coverage of handcrafted systems can be increased without the need to rebuild the linguistic tools by using machine learning techniques*. The hypothesis presented assumes the following:

- The effort of building a training set is less than that of building the linguistic resources.

- The learning algorithm will generalize well over errors performed by the handcrafted system.

The research work will provide a faster and easier method for adapting a handcrafted NE tagger to a new domain. The hypothesis presented here considers that it is possible to reduce the effort in portability to just gathering a training set for the learner. In addition, by eliminating the dependency on linguistic resources, the regular user will be capable of performing the adaptation task.

It is important then to note that even though in the handcrafted system there has no need for a training set, we consider the task of gathering this training set much faster and easier than that of building the linguistic resources of the handcrafted systems. Moreover, this task can be simplified to correct the output of a handcrafted system, instead of building the training set from scratch, which will not require the time nor the computational linguistic knowledge needed by the handcrafted systems.

The idea of using Machine Learning techniques for building NE taggers is not new, see [6,7,10,12,14]. However, in most of the previous works an important component of the system is a linguistic handcrafted resource. For example, Petasis et al. [12] proposed a combination of supervised and unsupervised techniques for adapting a proper noun dictionary to a new domain. Both techniques use as an initial knowledge source a list of proper nouns. Zhou and Su [7] reported good results when using gazetteer information as one of the features in a Hidden Markov Model; similarly Carreras et al. [10,14] and Arévalo et al. [6] use machine learning algorithms such as decision trees and AdaBoost in combination with lists of trigger words and gazetteers. Therefore, the resulting method of this research will be different from previous works since it will allow to improve performance of a NE tagger without requiring the revision of the manual linguistic resources on which the NE tagger is based.

There are two problems that need to be solved in order to achieve the goal of this research. The first involves the delimitation of an NE in text, to recognize a word or sequence of words as a probable NE and, of course, to discard any non-NE word. This is a difficult problem given that an NE can consist of one to any number of words. Once we have identified the limits of an NE, the second problem is to assign to it the appropriate label. These two problems can be solved separately, however possible NEs need to be recognized in order to perform NE classification.

At this time we have designed an initial algorithm for NE classification. This algorithm uses the output of a handcrafted system as one of the features. The preliminary results are encouraging as the method outperforms the handcrafted system. It is very likely that this method can be successfully applied to the problem of NE recognition, as well as to different languages, but this still needs to be validated experimentally.

3.1 Named Entities for Question Answering Systems

NEs are a very important source of information for many natural language applications such as Question Answering Systems. The solution proposed in this thesis will be validated in this direction by the development of a representation of documents as a set of NEs and their context. Preliminary results of this representation model to answer factual questions have shown that the model can help to improve the accuracy of such systems, as well as to sharpen the ability of giving concise answers. In order to improve performance of question answering systems further, the ability of automatically extracting relations among NEs is needed. If time allows research will be performed aimed at exploring new methods for automatic extraction of relations among NEs in text.

4. Methodology

Proving the correctness of the hypothesis stated in the previous section will require to perform several tasks, so far the authors have identified the following:

- *Collecting Corpora* It has been mentioned that one of the advantages of using machine learning techniques relies on the fact that no linguistic resources will be rebuild. As a practical alternative a training corpus will be employed by a learning algorithm. Then it is important to have at hand a corpus requiered by the training process but also helpful in the experiments. Besides this, at least two different corpora will be needed in order to provide evidence of portability of the proposed solution. One of these corpora is intended to be in a different language.
- *Acquiring NE taggers* Named Entity taggers are required in order to support the hypothesis. The goal is to use a Named Entity tagger for Spanish and if possible, other for a different language, perhaps English. The language of the second NE tagger depends on the availability of such systems.
- *Selecting a Machine Learning algorithm* The research of machine learning techniques will help in the selection of the algorithm for the proposed solution.
- *Assessing performance of the methods by experimentation* This includes experimenting with a corpus of a different domain and a corpus of a different language. The feasibility of performing experiments in a different language is restricted to the availability of a NE tagger for such language.
- *Evaluating results*
- *Validating the proposed method in Question Answering Systems*

4.1 Learning NE Classifiers

This section describes the algorithm that will be used as a starting point of this research project. In order to solve the problem of increasing the coverage of a Handcrafted Named Entity Tagger (HNET), this is posed as a learning problem where Support Vector Machines (SVM) [22] are used as the learning algorithm. The HNET is considered as a black box, the method is interested only in its outputs, which are used as attributes in the learning scenario. This proposed solution can be considered as a stack of classifiers where in the first stage, a traditional HNET is used to recognize NEs and assign possible tags to the corpus, then these tags are used by a SVM classifier to obtain the final NE tags. In addition, other attributes used are the class values assigned to the 2 words to the left and right of the instance, this size of window follows that of Petasis et al. [12], although we plan to perform experiments varying the size of the window. Preliminary experimental results show that Support Vector Machines are successfully applied to this learning task increasing F-measure and accuracy.

As mentioned previously, the NE classifier uses the output of a HNET. The authors believe that machine learning algorithms can be used to improve performance of handcrafted NE taggers without a considerable effort, as opposed to that involved in extending or redesigning grammars and lists of trigger words and gazetteers. Another assumption underlying this approach is considering that the misclassifications of the HNET will not affect the learner. Intuitively, the incorrectly classified instances can be considered as noisy. However by having available the correct NE classes in the training corpus, the learner will be capable of generalizing error patterns that will be used to assign the correct NE. If this assumption holds, learning from other's mistakes, the learner will end up outperforming the HNET.

In this preliminary work, the problem of recognizing which parts of the text are NEs is discarded. Named Entities are recognized and segmented by the HNET, the only concern being the NE classification in the same four types defined in the CoNLL 2002 competition: Person (*Per*), Location (*Loc*), Organization (*Org*) and Miscellaneous (*Misc*).

In order to build a training set for the learner, each instance n is described by a vector of six attributes, $\langle a_1, a_2, \ldots, a_6 \rangle$, where a_1 and a_2 are the classes assigned by the HNET to the first two words to the left of n, a_3 and a_4 are the classes for the first two words to the right of n, a_5 is the class value of n predicted by the HNET and a_6 is the true class value of n. Note that a_5 and a_6 will differ only when the base HNET misclassifies a named entity. There are two types of possible errors from the HNET that the learner will try to overcome. The first type occurs when the HNET assigns the wrong NE class to the NE; the other type occurs when the HNET fails to recognize the NE. When this is the case the HNET assigns a PoS tag to the NE, although the same thing happens when the HNET needs to classify a non-NE word. It follows that the possible values for attributes a_1 to a_5 can take any value from the set of possible NEs, this is $NE=\{Per, Loc, Org, Misc\}$, plus the set of possible PoS tags, which has over one hundred different tags; so the size of the feature space of our NE classification task is 5×10^{10}.

A graphical representation of the proposed NE classifier is given in Figure 1. It can be seen that a HNET is used as a black box and its output is fed to the SVM classifier. The SVM classifier uses this information (in some cases also additional information such as PoS tags, capitalization information and lemmas) and assigns the final NE classification.

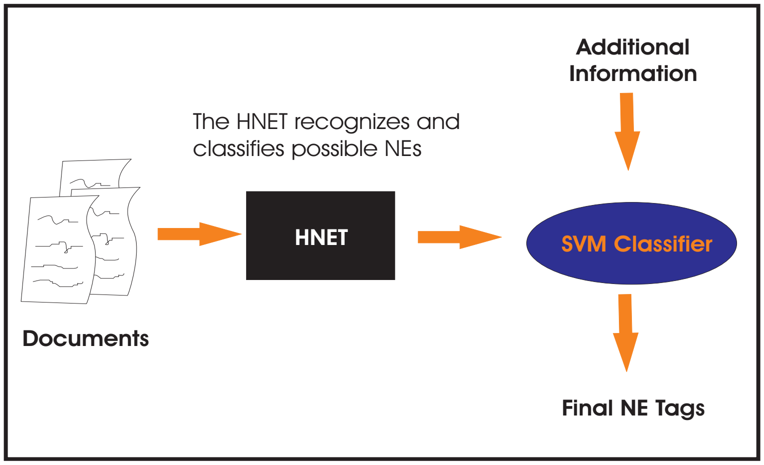

Figure 1. A graphical representation of our NE classifier. The possible NEs together with the tags given by the HNET are used as input to the SVM classifier.

4.2 Low-Dimensionality Features

Having the high-dimensionality feature space described above has some serious drawbacks: the number of training examples needed in order to achieve good accuracy is proportional to the size of the feature space; but increasing the amount of training examples to meet this requirement might be unfeasible. Another disadvantage is the high computational resources needed to use SVM with a feature space such as this.

In order to overcome these difficulties we decided to reduce the size of the feature space. We achieve this by generalizing the POS tags, i.e. instead of having tags for all the possible kinds of pronouns, we kept one tag *P* that encapsulates the set of pronouns. We did the same for the different POS categories. Table 1 shows the resulting reduced set of feature values. By doing this our reduced feature space has a size of $16^5 \times 4 = 4,194,304$.

Feature Value	Description
Per	Person
Org	Organization
Loc	Location
Misc	Miscelaneous
N	Noun
A	Adjective
P	Pronoun
F	Punctuation mark
D	Determiner
V	Verb
S	Preposition
R	Adverb
T	Article
C	Conjunction
M	Numeral
I	Interjection

Table 1. Reduced Feature Values Set

4.3 Preliminary Results

This section describes the experimental setting used to evaluate the algorithm presented previously. In this setting multi-class problems are solved using pairwise classification. The optimization algorithm used for training the support vector classifier is an implementation of Platt's sequential minimal optimization algorithm [23]. The kernel function used for mapping the input space was a polynomial of exponent one.

Two data sets were used in the experiments, one was gathered by people in the NLP lab at INAOE. It consists of news in Spanish acquired from different newspapers from Mexico that cover disaster-related events. This collection contains a total of 285 NEs. Although it is a small corpus (which is not so bad as it gives the opportunity of manual revision) it is large enough for the experimental evaluation. The other corpus is that used in the CoNLL 2002 competitions for the Spanish NE extraction task. This corpus is divided in three sets: a training set consisting of 20,308 NEs and two different sets for testing, *testa* which has 4,634 NEs and *testb* with 3,948 NEs, the former was designed to tune the parameters of the classifiers (development set), while *testb* was designed to compare the results of the competitors. As in our setting there is no parameter tuning, we performed experiments with the two sets.

Data Sets	Precision	Recall	F_1	Accuracy
Baseline	90.80	66.2	76.62	62.10
SVM Features	78.60	89.95	83.91	73.38
SVM Features+NEE	87.02	90.83	88.88	80.00
SVM NEE (*)	88.30	91.73	89.96	81.75
% of improvement (best vs baseline)	**-2.83**	**27.83**	**14.83**	**24.04**

Table 2. Results of NE categorization with the disaster data set. The last row shows the percentage of highest improvement comparing the best result with SVM, marked with "*", against the baseline.

In order to evaluate the proposed method, the HNET considered is that developed by Carreras and Padró [24]. They have developed a set of NLP analyzers for Spanish, English and Catalan that include practical tools such as PoS taggers, semantic analyzers and NE extractors. This HNET is based on hand-coded grammars and lists of trigger words and gazetteer information.

First the results using the disaster corpus will be discussed. Four different experiments were performed and the results are in Table 2. In all the experiments 10-fold cross-validation technique was used. The first experiment, labeled Baseline is the result of the HNET. As can be seen, it has very high precision, while recall and thus F-measure, are not so good. The Baseline system achieved an accuracy of 62.10%. A different experiment was performed using SVM trained on features like PoS tags, lemma and capitalization information. These features are acquired automatically from the text using the Spanish analyzers mentioned above. A five word window was used, where 2 words to the left and 2 words to the right of the target word are considered, each of these words is described by its PoS tag, its lemma and the capitalization of the word (all letters capitalized, first letter capitalized, digits or other when none of these is true). Each target word is then described by a vector of 16 attributes: 5 times 3 features plus the real class value. Results from this experiment are also in Table 2 under label SVM Features. Even though in this experiment the HNET tags are not used, the machine learning method achieves higher recall, F-measure and accuracy than the HNET.

The results named SVM Features+NEE were obtained using the same features described in the previous experiment plus the NE tags assigned by the extractor system. This combination of features outperformed the previous results in all but one figure, achieving an accuracy of 80%. The last experiment performed with this corpus uses as features the output of the HNET, it does not use PoS tags or additional information. Results are labeled SVM NEE, also shown in Table 1. The methods using SVM outperformed the HNET in recall, F-measure and accuracy, the best results being those from training SVM on the extractor system tags. Precision of the HNET was the only figure that remained higher. However the improvements achieved by using SVM are as high as 27% in recall, 14.83% in F-measure and 24.04% in accuracy.

Data Sets	Precision	Recall	F_1	Accuracy
Baseline	77.4	95.49	85.48	74.64
SVM Features+NEE (*)	86.00	92.27	88.97	80.14
SVM Features	67.00	83.02	74.12	58.89
SVM NEE	85.60	91.31	88.36	79.15
% of improvement (best vs baseline)	**10**	**-3.49**	**3.92**	**6.86**

Table 3. Results of NE categorization for CoNLL 2002 *development* set. The last row shows the percentage of highest improvement comparing the best result with SVM, marked with "*", against the baseline.

Data Sets	Precision	Recall	F_1	Accuracy
Baseline	70.50	88.08	78.33	64.38
SVM Features+NEE (*)	80.06	87.40	83.86	72.21
SVM Features	71.60	81.55	76.23	61.60
SVM NEE	80.20	86.57	83.28	71.35
% of improvement (best vs baseline)	**11.94**	**-0.78**	**6.59**	**10.84**

Table 4. Results of NE categorization for CoNLL 2002 *test* set. The last row shows the percentage of highest improvement comparing the best result with SVM, marked with "*", against the baseline.

Tables 3 and 4 present results using the corpora from the CoNLL 2002 competition. In both tables the designated training set was used to build the classifiers. The first table shows the results of using the development set for testing. It can be seen that higher precision, F-measure and accuracy can be achieved by using the HNET tags and additional features (PoS tags, capitalization and lemmas), while the HNET by itself has the highest recall for this set. Table 3 shows the results of testing with the test set of CoNLL 2002, the SVM Feature+HNET classifier outperformed the baseline method in three figures: precision, F-measure and accuracy.

By comparing the results from the three tables it is interesting to note that for the CoNLL 2002 test sets the best results are achieved by combining features such as PoS tags, capitalization and lemma information with the output of the HNET. While in the disaster data set the best results are achieved by using only the output of the extractor system. This is not surprising considering an important difference between the disaster corpus and those from the CoNLL 2002: the disaster-related corpus was carefully checked, so the NE tags are nearly error-free, while the other corpora are very large, which makes it unfeasible to manually correct any misclassification. There are some inconsistencies in these corpora that we were unable to correct. By using additional information, SVM can achieve better results when there may be noise in the examples, as it is suspected this is the case. However, the three learning tasks share a common characteristic, the average best results were achieved by a method using SVM. The NE extractor system was always outperformed by some method based on machine learning in at least three out of the four measures. These results were reported in the paper by Solorio and López López [25].

4.3 Discussion

The previous subsection presented an experimental evaluation of the proposed method that provides a promising line of research. However, in order to fulfill the thesis goal there is much more research work to do. Below is a "work to do" list containing several interesting ideas.

1. Exploring the use of ensemble methods. So far SVM have shown good results, but it will be interesting to explore if an ensemble of SVM can improve further the performance of the method.
2. Experimenting with documents of a different domain. It is a fundamental task of this research given that it will help support the hypothesis.
3. Exploring the use of the method in a different language. The portability of the system can be further corroborated if the method attains good results in a different language.
4. Validating the method in a Question Answering System. Pérez-Coutiño et al. present preliminary results of applying the proposed method in a document representation for Question Answering systems [26]. This approach will be further explored.
5. Exploring the use of the proposed solution to the problem of NE recognition. There is ongoing work for gathering a training corpus for extending the proposed method to NE recognition. There is the need for a handcrafted system for NE recognition. If it is not possible to acquire one then a training set will be used in order to build an automated NE classifier.

The research developed so far has brought out some very interesting open problems such as the use of unlabeled data in the problem of NE classification. Given that unlabeled data are more easily gathered, as opposed to labeled, there have been some interesting methods proposed in problems such as text categorization that exploit information contained in unlabeled data [27]. Using unlabeled data can benefit NE categorization and machine learning algorithms such as SVM can be an interesting alternative in this direction.

Another attractive open problem is the automatic correction of errors in corpus. A major difficulty of doing research related with natural language processing is the lack of error-free annotated corpora. It can be a very extenuating task to verify the correctness of the tagging. Machine learning can be used for building error detectors in corpora.

5. Possible Contributions

Now that the research problem has been stated, and the methodology to follow has been described, this section presents the possible contributions of this research.

When this research project be completed, we foresee for now the following contributions:

- *A new methodology for adapting the coverage of traditional handcrafted NE taggers for Spanish* This is the most ambitious goal in this research, as it implies that at the end of this research work we will be able to prove the correctness of the hypothesis stated in section 3. If the hypothesis of the research is successfully proved, then one important contribution is to acknowledge the fact that by using machine learning techniques, systems that perform poorly, due to limitations of the linguistic resources they rely on, can be automatically enriched without the tedious and complex task of redesigning those linguistic resources.
- *Portability of the method to a different language* The research work is focused primarily in performing experimental evaluations of the methods in Spanish. The reason for choosing Spanish is that there are far less research efforts for developing technology for this language

compared to other languages such as English. However, experimental work will be performed using English documents given that it is useful having the ability of porting NE taggers to different domains, but it is also an attractive feature being able to use the method for adapting the coverage of a NE tagger in a different language. It is important to clarify that it is not an objective of the thesis to design a method independent of the language. The objective is just to explore the possibility of using the method in English documents leaving a starting point for the more ambitious goal of a NE tagger independent of language.

- *Increasing resources for Natural Language Processing Systems for Spanish* Providing a method for accurate NE classification for Spanish represents a practical contribution toward developing sophisticated tools for Natural Language applications targeted to Spanish-speakers. Proposing a document model representation based on NEs for Question Answering systems is an example of the usefulness of such method.

References

[1] Andrew Borthwick. *A Maximum Entropy Approach to Named Entity Recognition*. PhD thesis, New York University, New York, September 1999.

[2] G. Mann. Fine-grained proper noun ontologies for question answering. In *SemaNet'02: Building and Using Semantic Networks*, 2002.

[3] Martin Hassel. Explotation of named entities in automatic text summarization for swedish. In *14th Nordic Conference of Computational Linguistics*, *NODALIDA 2003*, May 2003.

[4] Bogdan Babych and Anthony Hartley. Improving machine translation quality with automatic named entity recognition. In : *Proceedings of the EACL 2003 Workshop on MT and Other Language Technology Tools Improving MT through other language technology tools, Resources and tools for building MT*, pages 1-8, 2003.

[5] P. Velardi, P. Fabriani, and M. Missiko. Using text processing techniques to automatically enrich a domain ontology. In *Proceedings of the International Conference on Formal Ontology in Information Systems*, pages 270–284, ACM Press, 2001.

[6] M. Arévalo, L. Márquez, M.A. Martí L. Padró, and M. J. Simón. A proposal for wide-coverage spanish named entity recognition. *Sociedad Española para el Procesamiento del Lenguaje Natural*, 28, pages 63-80, 2002.

[7] G. Zhou and J. Su. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of ACL'02*, pages 473–480, 2002.

[8] R. Florian. Named entity recognition as a house of cards: Classifier stacking. In *Proceedings of CoNLL-2002*, pages 175–178, Taipei, Taiwan, 2002.

[9] T. Zhang and D. Johnson. A robust risk minimization based named entity recognition system. In *Proceedings of CoNLL-2003*, pages 204-207, Edmonton, Canada, 2003.

[10] X. Carreras, L. Márquez, and L. Padró. A simple named entity extractor using adaboost. In *Proceedings of CoNLL-2003*, pages 152-155, Edmonton, Canada, 2003.

[11] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of CoNLL-2003*, pages 188-191, Edmonton, Canada, 2003.

[12] G. Petasis, A. Cucchiarelli, P. Velardi, G. Paliouras, V. Karkaletsis, and C. D. Spyropoulos. Automatic adaptation of proper noun dictionaries through cooperation of machine learning and probabilistic methods. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 128–135, ACM Press, 2000.

[13] S. Sekine, K. Sudo and C. Nobata. Extended named entity hierarchy. In *Proceedings of the LREC 2002 Conference*, pages 1818-1824, Las Palmas, Canary Islands, Spain, 2002.

[14] X. Carreras, L. Márquez and L. Padró. Named entity extraction using adaboost. In *Procedeengs of CoNLL-2002*, pages 167-170, Taipei, Taiwan, 2002.

[15] T. Dietterich. Ensemble methods in machine learning. In *First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science*, pages 1–15, New York: Springer Verlag, 2000. In J. Kittler and F. Roli (Ed.).

[16] E. F. Tjong Kim Zhang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the CoNLL-2003*, pages 142-147, Edmonton, Canada, 2003.

[17] U. Hahn and K. G. Mark. Joint knowledge capture for grammars and ontologies. In *Proceedings of the International Conference on Knowledge Capture*, pages 68–75. ACM Press, 2001.

[18] S. Basu, R. J. Mooney, K. V. Pasupuleti, and J. Ghosh. Evaluating the novelty of text-mined rules using lexical knowledge. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 233–239, San Francisco, CA, 2001.

[19] B. Rosario and M. Hearst. Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy. In *Proceedings of 2001 Conference on Empirical Methods on Natural Language Processing, (EMNLP 2001)*, pages 82-90, Pittsburgh, PA, 2001.

[20] R. Girju, A. Badulescu, and D. Moldovan. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the Human Language Techonology Conference*, pages 80-87, Edmonton, Canada, 2003.

[21] D. Roth and W. Yih. Probabilistic reasoning for entity & relation recognition. In *COLING'02*, 2002.

[22] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995.

[23] J. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods –Support Vector Learning*, (B. Scholkopf, C. J. C. Burges, A. J. Smola, eds.), pages 185–208, Cambridge, Massachusetts, 1999. MIT Press.

[24] X. Carreras and L. Padró. A flexible distributed architecture for natural language analyzers. In *Proceedings of LREC'02*, Las Palmas de Gran Canaria, Spain, 2002.

[25] T. Solorio and A. López López. Learning named entity classifiers using support vector machines. In *A. Gelbukh (ed.), CICLing-2004, Lecture Notes in Computer Science 2945: Computational Linguistics and Intelligent Text Processing*, pages 158-166, Springer-Verlag, 2004.

[26] M. Pérez-Coutiño, T. Solorio, M. Montes y Gómez, A. López-López, and L. Villasenor-Pineda. Toward a document model for question answering systems. In *Atlantic Web Intelligence Conference AWIC04, Lecture Notes in Computer Science*. Springer-Verlag, 2004. (to appear).

[27] K. Nigam, A. Mc Callum, S. Thrun, and T. Mitchell. Learning to classify text from labeled and unlabeled documents. *Machine Learning*, pages 1–22, 1999.