eXtended WordNet: progress report

Rada Mihalcea and **Dan I. Moldovan**
Southern Methodist University
Dallas, TX, 75275-0122,
{rada,moldovan}@engr.smu.edu

Abstract

eXtended WordNet (XWN), a morphologically and semantically enhanced version of the WordNet dictionary, is currently build at SMU [1]. There are several phases in the XWN project. This paper focuses on the semantic disambiguation stage of this project, and the preprocessing required by this stage.

1 Introduction

WordNet became lately one of the most frequently used machine readable dictionaries, its popularity being mostly due to the rich set of semantic relations it encodes. WordNets in other languages started to become available as well, as part of the European WordNet project or similar projects.

There is a large range of applications using WordNet, including word sense disambiguation, knowledge acquisition, text inference, information retrieval, conceptual indexing, question answering and others. Many WordNet applications require additional semantic and logic enhancements that the current WordNet does not have. This need has motivated the XWN project.

The XWN project involves several processing phases applied to the glosses. We concentrate in this paper on the semantic disambiguation stage that aims at assigning sense tags to all the words in the glosses. It requires a preprocessing phase as well, including part of speech tagging and concept identification.

Another issue addressed in this paper is that of automatic verification of tagging accuracy. Our goal is to develop tools that enable automatic labeling of the words in the glosses with morphological and semantic tags. On the other hand, one of the main requirements of the XWN project is correctness, and this is actually one of the hardest to achieve. We have to find a viable compromise between the goal of automatic labeling and the requirement of high precision. Section 2 summarizes the theory related to the assignment of levels of confidence in tagging corpora, when two or more taggers are employed.

1.1 eXtended WordNet

The eXtended WordNet was introduced for the first time in (Harabagiu et al., 1999). The main transformations aimed by this project refer to the WordNet glosses and the semantic and logic relations that can be derived from these glosses. There are four main phases proposed in this project:

1. Preprocessing and parsing. This stage implies also the separation of glosses into definitions and examples, tokenization and the identification of compound words.
2. Word sense disambiguation (WSD). All the open class words in the glosses will be tagged with the appropriate senses from WordNet. The words will be linked to their corresponding synsets, and therefore this step will allow for the derivation of topical relations (step 4).
3. Logical form transformation. The glosses are transformed into logical forms enabling applications such as text inference or axiomatic proofs.
4. Topical relations. A larger set of connections can be established among words, independent of their part of speech, based on their association with a particular context or topic. Such relations are useful for information retrieval, information extraction, text coherence and other applications.

In this paper, we focus on the work we started to do for the first two stages, namely preprocessing and word sense disambiguation of WordNet glosses. We detail the methods we plan to employ for these tasks and exemplify the usage of the tools designed for each of these stages in processing a set of 1000 verb and noun glosses [2]

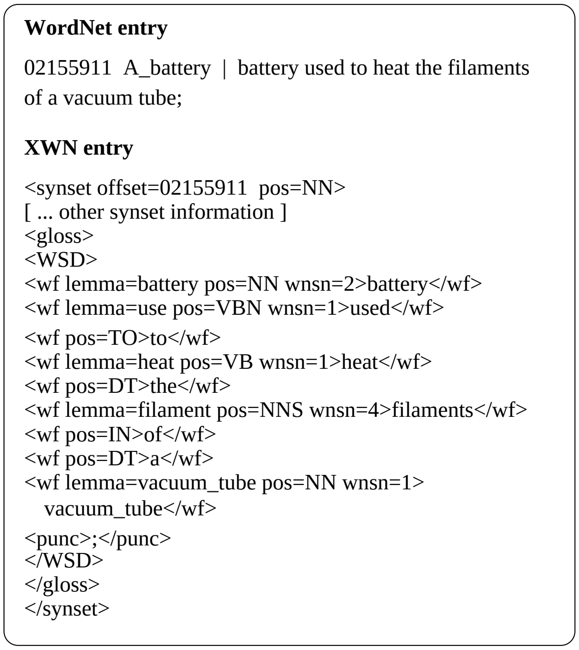
1.2 XWN format

One of the issues addressed from the very beginning and discussed together with the Princeton team, was

[1] This research has been supported in part by NSF grant EIA-0078854

[2] As a first step of this project, we decided together with the Princeton team to start by processing one noun class and one verb class, respectively the *noun.artifact* and *verb.social* classes. For the experiments reported in this paper, we used a subset of 1000 glosses extracted from these hierarchies.

the XWN database format. The most important requirements concerning the format are flexibility and scalability: the notation chosen should allow for the incorporation of future information without affecting the current settings. We have agreed on the following format, which makes use of SGML tags and is similar with the notation used in SemCor:
- each word should include a part of speech tag;
- words defined in WordNet should include a lemma (i.e. word baseform) and sense field;
- punctuation has to be marked accordingly;
- the various processing stages are to be separated using specific tags.



WordNet entry

02155911 A_battery | battery used to heat the filaments of a vacuum tube;

XWN entry

```
<synset offset=02155911  pos=NN>
[ ... other synset information ]
<gloss>
<WSD>
<wf lemma=battery pos=NN wnsn=2>battery</wf>
<wf lemma=use pos=VBN wnsn=1>used</wf>
<wf pos=TO>to</wf>
<wf lemma=heat pos=VB wnsn=1>heat</wf>
<wf pos=DT>the</wf>
<wf lemma=filament pos=NNS wnsn=4>filaments</wf>
<wf pos=IN>of</wf>
<wf pos=DT>a</wf>
<wf lemma=vacuum_tube pos=NN wnsn=1>
   vacuum_tube</wf>
<punc>;</punc>
</WSD>
</gloss>
</synset>
```

Figure 1: An example of XWN format

Figure 1 shows an example of a gloss obtained from the disambiguation stage. Note that only the information related to the semantic disambiguation process is shown here, delimited by the <WSD> tags. Other phases will have separate sections delimited with tags such as <LFT> (for logic form transformations) or <TR> (for topical relations).

2 Levels of confidence in building tagged corpora

In this section, we summarize the results reported in (Mihalcea and Bunescu, 2000) regarding the assignment of levels of confidence in building tagged corpora. The idea of combining several classifiers for the purpose of achieving a better accuracy is not new, but it was never adapted before to the task of verifying tagging correctness. The idea presented in their report, as well as the formalization of the lower and upper bounds for the precision that can be achieved when combining several classifiers, fits well with our project and its requirements of high accuracy.

Given two taggers, denoted with T_1 and T_2, and their precision estimates P_{T_1} and P_{T_2}, if the two taggers agree in a number of cases denoted with *cov*, it is possible to find a lower and upper bound of the precision obtained on the set where the two taggers agree as a function of the accuracies of the individual taggers and the size *cov* of the agreement set. Denoting the precision achieved on this agreement set with P_{cov}, the following bounds are determined:

$$minP_{cov} = \frac{P_{T_1} + P_{T_2} - 1 + cov}{2 * cov} \quad (1)$$

$$maxP_{cov} \simeq \frac{P_{T_1} + P_{T_2} - 1 + cov + (1 - P_{T_1})(1 - P_{T_1})}{2 * cov} \quad (2)$$

Experiments presented in their report show the validity of these theoretical results. A generalization of these equations to more than two taggers is also provided.

We find these formulae very useful for our task of determining the correctness of morphological and semantic tagging. Basically, given two or more taggers, and the number of cases where the two taggers agree, one can determine the precision of the tagging correctness on the agreement set. If this accuracy is high enough for the proposed task, then a human will check only the rest of the cases where the taggers disagree. More than that, if the precision is not satisfactory, then three or more taggers can be combined until the desired accuracy is achieved, even if the agreement set is smaller. The words tagged differently by the taggers involved will be checked by a human.

3 Part of speech tagging

For the purpose of disambiguating the words in glosses, it is necessary to have a certain level of preprocessing information, including part of speech tagging and compound words identification. Since it is one of the first steps, the accuracy achieved in this stage directly impacts the rest of the text processing steps.

The state-of-the art tools in part of speech tagging have a reported accuracy of about 93-94%. Although this figure is very high, it might not be enough given the fact that a wrong part of speech definitely leads to a wrong semantic tag.

To solve this problem, we used the result reported in (Mihalcea and Bunescu, 2000) to determine a combination of part of speech taggers that would provide a minimum accuracy of 98% on the agreement set. This way, a human has to check only the cases where the two taggers disagree, such as to obtain an overall accuracy closed to 100%.

We had three taggers available, namely a rule based part of speech tagger (Brill, 1992), a probabilistic tagger Qtag (Mason, 1997) and Mxpost, a tagger based on the maximum entropy principle (Ratnaparkhi, 1996). The reasons for choosing these taggers are (1) their public availability, (2) their accuracy and (3) the set of tags used. The estimated accuracies of these taggers are 94% for Brill tagger, 87% for Qtag and 96% for Mxpost. The agreement set (i.e. the number of cases when two taggers agree) was measured as 85% when Brill tagger and Qtag are used, respectively 94% when Mxpost and Brill tagger are employed. Using the formula for the minimum precision on the agreement set given by equation 1, it results that the minimum accuracy that can be achieved for these combinations of taggers is:

$$minP_{Brill+Qtag} = \frac{0.94 + 0.87 - 1 + 0.85}{2 * 0.85} = 0.976 \qquad (3)$$

$$minP_{Brill+Mxpost} = \frac{0.94 + 0.96 - 1 + 0.94}{2 * 0.94} = 0.978 \qquad (4)$$

It is interesting to observe that a combination of more precise taggers does not necessarily provide higher combined accuracy. In fact, the two tagger combinations have the same minimum accuracy, but we decided to use the second combination (Mxpost and Brill's tagger) since it provides a larger agreement set (94%).

Besides part of speech tagging, we also need to identify the compound words, based on WordNet definitions. These compound words are identified automatically based on the principle of the longest succession of words defined in WordNet. As we did not find yet an automatic way of validating the correctness of this process, a human checks whether the automatically extracted concepts are correct or not.

A tool, called *xwnPreprocess*, combines all these functionalities required by the preprocessing phase, and allows for human intervention when needed. The functions performed by *xwnPreprocess* are summarized in Figure 2.

Table 1 summarizes the part of speech information gathered from 1000 glosses, as well as the number of compound words identified automatically and the number of compound words considered correct by a human.

Time considerations As illustrated in figure 2, the user has to interact with the preprocessing system in two cases: (1) to indicate the correct part of speech when there is a mismatch among the automatic taggers involved and (2) to check the correctness of the automatically extracted compound words. The time spent for validating the part of speech tagging of the 8659 words and the compound concepts in the 1000 glosses was four hours. This is worthwhile considering that a performance of nearly 100% was achieved.

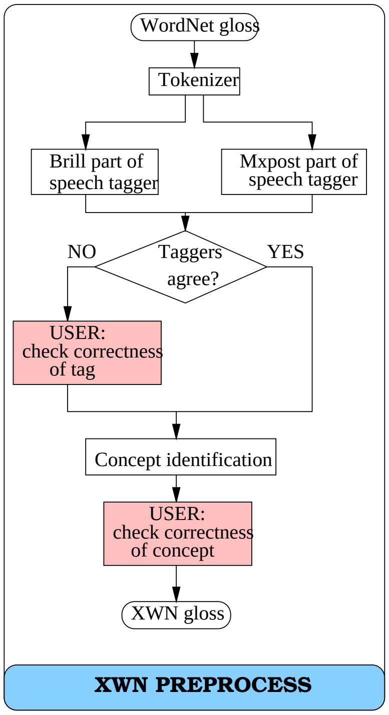

WordNet gloss
Tokenizer
Brill part of speech tagger
Mxpost part of speech tagger
NO
Taggers agree?
YES
USER: check correctness of tag
Concept identification
USER: check correctness of concept
XWN gloss
XWN PREPROCESS

Figure 2: The XWN tools

No.	Noun glosses	Verb glosses
Words	5539	3120
Nouns	1699	773
Verbs	758	719
Adjectives	381	181
Adverbs	95	112
Concepts		
identified	413	191
correct	232	104

Table 1: Part of speech tagging and compound words identification on 1000 glosses

4 Semantic disambiguation of glosses

Continuous effort has been made towards the goal of solving the problem of semantic ambiguity, including the Senseval competition (Kilgarriff and Rosenzweig, 2000), aiming at evaluating existing WSD methods. The overall conclusion is that no single method can provide a complete solution to this problem, but a battery of procedures is needed to distinguish among word senses.

The procedures described in this section combine new and old techniques for semantic disambiguation, bringing together heuristics, corpus evidence and distances computed on semantic nets. The ultimate goal is to disambiguate all the words in the WordNet glosses with a very high precision. The requirement of 100% recall and over 90% precision is very hard to achieve with completely automatic tools, given the state of the art in this field, which does not go over 70-80% precision.

The semantic disambiguation of glosses is slightly different with respect to the task of solving the semantic ambiguity of words in open text, as we know the concept to which a gloss belongs to and the gloss disambiguation can benefit from this fact.

We present here the procedures we have used and implemented so far and present the results obtained with these procedures.

The input to this module, called *xwnWsd*, is constituted by a gloss in XWN format, as obtained with the *xwnPreprocess* tool. In *xwnWsd* we count on a correct part of speech tagging and a correct concept identification.

4.1 Checking correctness for WSD

In section 2, we have presented a scheme that enables the association of levels of confidence with the correctness of the labels assigned to the words in a corpus. Using equation 1, we can determine the minimum accuracy of a combination of taggers; if this accuracy is high enough, then we can check the tags only for the set where the taggers disagree. Otherwise, more taggers can be employed until we reach the desired level of precision.

We plan to use these theoretical results in assigning levels of confidence for WSD as well. The problem in this stage of the project is that we do not have yet available at least two different complete disambiguation methods that would address all the words in the glosses, such as to be able to combine them in a manner similar with the morphological tagging. Instead, we have several methods that address subsets of words, and label them with semantic tags based on various constraints. As these methods are not complete yet, we need a scheme that takes into account this fact and determines the precision and correspondingly assigns levels of confidence to subsets of words, rather than to the whole text.

Based on the tagging precision of each method and/or combinations of methods, we can assign a numerical level of confidence to each word in the range 0-2, with the following meaning:
0 - the word was not addressed by any method;
1 - the word was addressed by one or two methods, but it has a tagging accuracy lower than 90%;
2 - the word has a tagging accuracy over 90%, due to one single very accurate method (such as the monosemous words procedure), or due to the agreement of two methods.

Practically, only the words having a level of confidence 2 will be considered correct; the other words have to be checked by a human. The problem becomes now how to combine the various methods; to solve this problem, we can make use of equation 1.

4.2 WSD procedures

We present in this section various procedures designed and implemented so far for the disambiguation of words in glosses.

4.2.1 Monosemous words.

Identify the words having only one sense in WordNet, and mark them with the appropriate sense.
Example. The gloss for *abbey#3* is *"a monastery ruled by an abbot"*. The word *abbot* has only one sense in WordNet, thus it is not ambiguous and we label it with sense #1.

4.2.2 Same hierarchy relation.

Identify the gloss words belonging to the same hierarchy as the synset of the gloss. This procedure was for the first time proposed in (Harabagiu et al., 1999) and it referred to the head word of the gloss. We generalize it to all the words in the gloss, and as shown later on in this paper, this generalization does not harm the precision of this method.
Example. The gloss for *devolve#1* is *"pass on or delegate to another"*. *delegate#2* belongs to its hypernym synset, thus this verb is disambiguated and labeled with sense *#2*.

4.2.3 Lexical parallelism relation

Identify the gloss words involved in a parallel relation. This is again a procedure previously presented in (Harabagiu et al., 1999). As we are not previously parsing the gloss, we determine these parallel relations simply as pairs of words separated by a conjunction or by a comma. These words should belong to the same hierarchy, in the case of nouns and verbs, or to the same cluster, if they are adjectives or adverbs. If one of the words in such a pair is already disambiguated by one of the previously applied methods, then the sense of the disambiguated word constitutes a restriction over the possible combinations of senses for the two words.
Example 1. The gloss for *aba#2* is *"a fabric woven from goat and camel hair"*. The two words here that are lexically parallel are *goat* and *camel*; *camel* is already semantically disambiguated, due to procedure 4.2.1, and the only sense of *goat* belonging to the same hierarchy as *camel#1* is *goat#1*.
Example 2. The gloss for *exert#3* is *"make a great effort at a mental or physical task"*. *mental* and *physical* are both ambiguous here, but there is an adjective cluster where they both belong to, with their senses *mental#1* and *physical#1*.

4.2.4 SemCor bigrams

We can benefit from the information that can be gathered from SemCor (a corpus tagged with the WordNet senses). With this procedure, we are trying to get contextual clues regarding the usage of a word sense. For a given word W_i, at position i in the gloss, form two pairs, one with the word before W_i

(pair W_{i-1}-W_i) and one with the word after W_i (pair W_i-W_{i+1}). Determiners or conjunctions cannot be part of these pairs. Then, we search for all the occurrences of these pairs found within SemCor. If, in all the occurrences, the word W_i has only one sense #k, and the number of occurrences of this sense is larger than a given threshold, then mark the word W_i as having sense #k.

Example. Consider the word *approval* in the text fragment *"committee approval of"*, and the threshold set to 3. The pairs formed are *"committee approval"* and *"approval of"*. No occurrences of the first pair are found in the corpus. Instead, there are four occurrences of the second pair:

" with the *approval#1 of* the Credit Association"
"subject to the *approval#1 of* the Secretary of State"
"administrative *approval#1 of* the reclassification"
"recommended *approval#1 of* the 1-A classification"

In all these occurrences *"approval"* has sense *#1*, and we label it accordingly.

4.2.5 Cross reference

Given an ambiguous word W in a gloss G belonging to the synset S, find if there is a reference from the definition of one of the senses of W to the words in the synset S. If there is such a relation, this is considered to be a cross reference between the words in S and the particular sense of W, and this sense is picked as correct.

Example 1. The synset {*agora#3, forum#3, public_square#2*} is *"a place of assembly for the people in ancient Greece"*. Sense *#14* for *place* is *"a public square with room for pedestrians"*, and thus there is a cross reference between the gloss of *place#14* and *agora#3.*

Example 2. The gloss for the synset {*alarm_clock#1, alarm#4*} is *"wakes sleeper at preset time"*. There are 10 possible senses of time, and its sixth sense is defined as *"the time as given by a clock"*. Again, we find a cross reference between this definition of time, and the definition of alarm, and we can label *time* with sense *#6.*

4.2.6 Reversed cross reference

Given an ambiguous word W in a gloss G belonging to the synset S, find if there is a reference from the definition G to a word in one of the synsets of the various senses of W. If there is such word, this is considered to be a reversed cross reference between the current gloss and the synset of that particular sense of W, and this sense is picked as correct.

Example 1. The gloss of *start#10* is *"begin work or acting in a certain capacity, office or job;"*. The noun *work* has 7 possible senses; the synset of its fourth sense is { *job, employment, work*} which has a reversed cross reference with the current gloss.

Example 2. The gloss for *withdraw#2* is *"withdraw from active participation"*. From the 17 possible senses of *active*, its fourth sense belongs to the synset {*active, participating*}, and this is the sense selected as correct, due to its reversed cross reference with the current gloss.

4.2.7 Distance among glosses

Given an ambiguous word W in a gloss G, we determine the number of common words among the glosses attached to its various senses and the words in the gloss G. This is a variant of the algorithm proposed by (Lesk, 1986) for the disambiguation of words in open text. We find it useful for our task; it basically gives a measure of the density among the current gloss and the possible senses of the ambiguous words in the gloss. There are cases when several senses of a word have the same number of common concepts with the current gloss; this is considered to be a tie, and we do not attempt to break this tie by randomly choosing a sense, but rather leave the word to be disambiguated by other methods.

When counting common concepts among definitions, we do not want to consider common words, like *the* or *use*; more than that, we would like to go across morphological classes, e.g. the words *support*, *supportive* and *supporting* should be determined as equivalent. In the current implementation, we are using the list of 572 common words provided with the SMART retrieval system and we stem the words using Porter stemmer (Porter, 1980) in an attempt to obtain the root of the word[3].

Example 1. Consider the example given in Figure 1. The noun *filament* has four different senses, but only sense #4 has the word *heat* in common with the given gloss.

Example 2. abacus#1 has the gloss *"a tablet placed horizontally on top of the capital of a column as an aid in supporting the architrave"*. For the word *capital*, ambiguous, its fifth sense has the definition *"the upper part of a column that supports the entablature"*, with two words in common with the current gloss (*support* and *column*) and this is the sense picked as correct by this method; *architrave* is also disambiguated using this procedure, with sense #2.

4.2.8 Common domain

There are cases when words can be disambiguated based on their domain. For example, the gloss for *mental#3* is *"(biology) of or relating to the chin- or lip-like structure in insects and certain mollusks"*. In this definition, *insect* is ambiguous, but from the two senses it has, only one relates to the (biology) domain, and this is the sense selected by this procedure.

[3] A different solution would be to use the functions from WordNet that extract the baseform of a word; the problem with this alternative is that it does not cross morphological classes, i.e. *supportive* and *support* will be still considered different words.

We did not implement yet this method, as the association of a synset with a particular domain is not yet determined for all synsets in WordNet. A simple inheritance algorithm will enable us to establish the domain information for a large set of synsets: once a domain is established for a particular synset, it propagates to all its hyponyms. For example, once we set the domain for the {*animal#1*} synset to be (biology), all its hyponyms (approximatively 7000 synsets) will inherit this feature, including {*insect#1*}.

Out of the eight procedures presented in this section, we have implemented and tested the first seven. Table 2 shows the recall and precision obtained for each of these procedures.

Procedure	Recall	Precision
P.4.2.1	21.3%	100%
P.4.2.2	13.2%	99%
P.4.2.3	11.9%	85.7%
P.4.2.4	16.2%	92.2%
P.4.2.5	4.2%	80%
P.4.2.6	5%	79%
P.4.2.7	17.9%	89.2%

Table 2: Word sense disambiguation on 1000 glosses

4.3 Combining the WSD procedures

Once we know the precision of the methods and their recall, we can analyze possible combinations of procedures that would provide the best performance. We do this by measuring the coverage between methods (i.e. the number of words tagged the same by two methods), and then determine the precision achieved on this coverage set using equation 1. Table 3 shows some possible combinations of methods, together with the coverage set and combined precision on the coverage set. The first two methods, namely 4.2.1 and 4.2.2 are not combined with any other methods, as they provide by themselves a high enough accuracy.

Proc.	P.4.2.3	P.4.2.4	P.4.2.7	P.4.2.5	P.4.2.6
P.4.2.3	-	80/96	80/98	-	-
P.4.2.4	-	-	85/95	74/98	71/99
P.4.2.7	-	-	-	74/96	69/99
P.4.2.5	-	-	-	-	66/94

Table 3: Values for *cov* (coverage) and $minP_{cov}$ (the minimum precision on the coverage set) for combinations of several methods.

As seen from Table 3, any combination of two methods provide a precision of over 90%. Depending on the level of accuracy required, we can use those combinations with a minimum of 90%, 95% or 98% precision. Future methods can be easily integrated into this scheme: for any method developed we need to determine its individual precision, size of the agreement set *cov* and precision on this set for the possible combinations with other methods. Based on this, the best combinations providing the minimum required accuracy will be chosen.

The partial result we achieved can be compared with the base line, obtained when assigning to each word its most frequent sense. On the 1000 glosses, this simple method resulted in 60% accuracy, much lower than the results we obtained.

5 Conclusion

The XWN project is work in progress. The task of automatically labeling words with morphological and semantic tags is very difficult, given the high accuracy required in this project.

The main result achieved so far is setting up a framework for the development of this task. First of all, we solved the problem of preprocessing, as required by the disambiguation stage: we implemented a tool that combines several taggers and allows for human intervention, such that the final outcome is *close to 100%* accuracy in part of speech tagging and compound words identification.

Next, we proposed several accurate procedures that enable automatic disambiguation of words. Even more important is the fact that we set up the methodology of combining procedures for best performance. Methods that will be developed in the future can be easily integrated and analyzed in this framework.

Another positive effect of the work we performed so far is that we have manually checked the correctness of the disambiguation process for 1000 glosses. This will constitute our *sense tagged corpus* to be used in testing future methods for semantic disambiguation.

References

E. Brill. 1992. A simple rule-based part of speech tagger. In *Proceedings of the 3rd Conference on Applied Natural Language Processing*, Trento, Italy.

S. Harabagiu, G. Miller, and D. Moldovan. 1999. WordNet 2 - a morphologically and semantically enhanced resource. In *Proceedings of SIGLEX-99*, pages 1–8, Univ. of Maryland.

A. Kilgarriff and R. Rosenzweig. 2000. English SENSEVAL: Report and results. In *Proceedings of LREC 2000*, Athens, Greece, June.

M.E. Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the SIGDOC Conference 1986*, Toronto, June.

O. Mason. 1997. QTAG-a portable probabilistic tagger. available online at: *http://www-clg.bham.ac.uk/QTAG/*.

R. Mihalcea and R. Bunescu. 2000. Levels of confidence in building a tagged corpus.

M. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods in Natural Language Processing Conference*, pages 130–142, Philadelphia, May.