
AutoASC — A SYSTEM FOR AUTOMATIC ACQUISITION OF SENSE TAGGED CORPORA

RADA MIHALCEA and DAN I. MOLDOVAN

*Department of Computer Science and Engineering
Southern Methodist University, Dallas, Texas, 75275-0122, USA
E-mail: {rada, moldovan}@seas.smu.edu*

Many natural language processing tasks, such as word sense disambiguation, knowledge acquisition, information retrieval, use semantically tagged corpora. Till recently, these corpus-based systems relied on text manually annotated with semantic tags; but the massive human intervention in this process has become a serious impediment in building robust systems. In this paper, we present AutoASC, a system which automatically acquires sense tagged corpora. It is based on (1) the information provided in WordNet, particularly the word definitions found within the glosses and (2) the information gathered from Internet using existing search engines. The system was tested on a set of 46 concepts, for which 2071 example sentences have been acquired; for these, a precision of 87% was observed.

Keywords: Natural language processing, corpora acquisition, sense tagged corpora, word sense disambiguation.

1. INTRODUCTION

Several corpus-based systems developed in the field of Natural Language Processing (NLP), performing tasks such as Word Sense Disambiguation (WSD), Knowledge Acquisition (KA) or Information Retrieval (IR), rely on semantically tagged text. These systems generally use the corpora to derive statistic information. As expected, the accuracy of these systems varies with the size of the available tagged text. So far, the annotation of the text with semantic tags has been done manually, which is expensive and time consuming. This impediment has limited the development of large tagged corpora.

In this paper, we present a system which automatically acquires sense tagged corpora. The main idea of this system was first described in Ref. 9. It is based on (1) the information provided in WordNet, particularly the word definitions found within the glosses, and (2) the information gathered from the Internet using existing search engines. Given a word for which a corpus is to be acquired, we first determine the possible senses that the word might have based on the WordNet dictionary. Then, for each possible sense, we either determine a *monosemous synonym* (this term is defined later in the paper), if such a synonym exists, or extract and parse the gloss specified in WordNet, if a monosemous synonym does not exist. Each gloss contains a *definition*, which can be used as a more detailed explanation for each sense of the word. The monosemous synonym or the definition is the base for a *search phrase*, used to search on the Internet. From the texts we gather, only those sentences containing the search phrase are selected. Next, the search phrase is replaced by the original word. In this way, we create example sentences for the usage of each sense of a word.

The idea of using the definitions is based on the fact that, in order to identify possible examples in which a particular sense of a word might occur, we need to locate that particular meaning of the word within some text. The definitions provided in WordNet are specific enough to uniquely determine each sense of the word, thus searching for these definitions will enable us to find concrete examples.

1.1. Motivation

One of the most important problems in the field of NLP, which can greatly benefit from an automatic method for semantic corpora acquisition, is the WSD problem.

Thus far, statistical methods have been considered the best techniques in WSD.⁷ They produce high accuracy results for a small number of preselected words; the disambiguation process is based on the probability that a word has a particular sense, given the context in which it occurs. The context is determined by the part of speech of each of the surrounding words, keywords, syntactic relations, collocations, etc. These disambiguation methods usually consist of two steps: (1) a first training step, in which rules are acquired using context information and (2) a testing phase in which the rules gathered in the first step are used to determine the most probable sense for a particular word. The weakness of these methods is the lack of widely available semantically tagged corpora.

The larger the corpora, the better the disambiguation accuracy. Typically, 1000–2500 occurrences of each word are manually tagged in order to create a corpus; from this, about 75% of the occurrences are used for the training phase and the remaining 25% are used for the test phase. Although high accuracy can be achieved with these approaches, a huge amount of work is necessary to manually tag words in the corpora.

For the disambiguation of the noun *interest* with an accuracy of 78%, as reported in Ref. 3, 2476 usages of *interest* were manually assigned with sense tags from the Longman Dictionary of Contemporary English (LDOCE).

For the LEXAS system, described in Ref. 14, the high accuracy is due in part to the use of a large corpora. For this system, 192,800 word occurrences have been manually tagged with senses from WordNet; the set consists of the 191 most frequently occurring nouns and verbs. As specified in their paper, approximately one man-year of effort was spent in tagging the data set.

To our knowledge, the only semantically tagged corpora with senses from WordNet is SemCor,¹¹ which consists of files taken from the Brown corpus. In SemCor, all the nouns, verbs, adjectives and adverbs defined in WordNet are sense tagged. Although SemCor is a large collection of tagged data, the information provided by SemCor is not sufficient for the purpose of disambiguating words with statistical methods.

Consider, for example, the noun *plane*, which has five senses defined in WordNet. The total of 39 occurrences which can be found in the SemCor files, for all the senses of this noun, is by far insufficient to create rules leading to high accuracy disambiguation results. Using the system described in this paper, we were able to

acquire 7010 examples for the different senses of the same word *plane*, thus more than 100 times than those found in SemCor.

There are also other problems in NLP which can benefit from the method described here. Some of the systems developed for KA need domain-specific corpora, which are usually created manually. Given some input words, the method presented here can be used to acquire specific examples. For example, a KA system in the financial domain might require corpora related to the words *interest#4* and *charge#8*. Obtaining these corpora can be easily accomplished with the system described in this paper. The usefulness of this method for some of the IR systems is straightforward. Our system can be used to retrieve information related to given input keywords.

1.2. Background on Resources

Several resources have been used in developing and testing our method. The first major step of extracting monosemous synonyms or definitions for each sense of a word is performed based on the information provided in WordNet. The second step, i.e. fetching examples from the Internet, makes use of the AltaVista search engine.

WordNet (WordNet 1.6 has been used in our method) is a Machine Readable Dictionary developed at Princeton University by a group led by George Miller.^{4,10} WordNet covers the vast majority of nouns, verbs, adjectives and adverbs from the English language. It has a large network of 129,504 words, organized in 98,548 synonym sets, called *synsets*. There is a rich set of 299,711 relation links among words, between words and synsets, and between synsets.⁶

The main semantic relation defined in WordNet is the “*is a*” relation; each concept subsumes more specific concepts, called *hyponyms*, and it is subsumed by more general concepts, called *hypernyms*. For example, the concept {**machine**} has the hypernym {**device**}, and one of its hyponyms is {**calculator**, **calculating machine**}.

WordNet defines one or more senses for each word. Depending on the number of senses it has, a word can be (1) *monosemous*, i.e. it has only one sense, as for example, the noun **interestingness** or (2) *polysemous*, i.e. it has two or more senses, as for example, the noun **interest** which has seven senses defined in WordNet.

The glosses in WordNet represent an important source of information. Almost all the synsets in WordNet have defining glosses. A gloss consists of a definition, comments and examples. For example, the gloss of the synset {**interest**, **interestingness**} is (the power of attracting or holding one’s interest (because it is unusual or exciting etc.); ‘they said nothing of great interest’; ‘primary colors can add interest to a room’). It has a definition: the power of attracting or holding one’s interest, a comment: because it is unusual or exciting etc. and two examples: they said nothing of great interest and primary colors can add interest to a room. Some glosses contain multiple definitions or multiple comments.

AltaVista,¹ is a search engine developed in 1995 by the Digital Equipment Corporation in its Palo Alto research labs, and is one of the most powerful search engines. In choosing AltaVista for use in our system, we based our decision on the size of the Internet information that can be accessed through AltaVista (it has a growing index of over 160,000,000 unique World Wide Web pages) and on the possibility to create complex search queries using Boolean operators: *AND*, *OR*, *NOT* and *NEAR*. The strongest constraint is imposed by *NEAR*: words connected with this operator have to be within a maximum distance of ten words from each other. From this point of view, *AND* is the weakest operator, words connected with *AND* have to be in the same document, without any constraints regarding the distance between them.

2. THE ARCHITECTURE OF AutoASC

In this section, we present the architecture and the procedures used by AutoASC, a system which enables the automatic acquisition of sense tagged corpora. The **input** to this system consists of a word for which example sentences are to be acquired for each of its different meanings. The raw corpus on which the system searches for these examples is formed by the largest available collection of texts electronically stored, namely the Internet. The **output** of the system is a set of sentences in which the original word is sense tagged.

The basic idea is to determine a lexical phrase, formed by one or several words, which uniquely identifies the meaning of the word, and then find examples including this lexical phrase. Such a lexical phrase can be created either using monosemous synonyms of the word or using the definition provided within the gloss attached to the WordNet synset in which the word occurs.

The system operation has three main phases: (1) **preprocessing** phase in which the gloss attached to the synset of a word $W\#i$ is parsed; (2) **search** phase in which search phrases are created using the procedures $P1$ through $P5$ described below, after which example sentences are sought on the Internet; (3) **postprocessing** phase in which the part of speech of the search phrase within the sentences gathered at Step 2 is checked to be the same as for the original word $W\#i$; the sentences in which the search phrases have the same part of speech as $W\#i$ become valid examples by replacing the search phrase with $W\#i$.

Figure 1 shows the architecture of AutoASC and identifies the three phases in the system execution.

2.1. Preprocessing

The main functionality of the **preprocessing** phase is to parse the gloss attached to a word synset. The parsing is done in six steps: the input to the parser is a gloss; the output is a set of definitions that are a part of speech tagged and phrase parsed.

Step 1. From each gloss, extract the definition part.

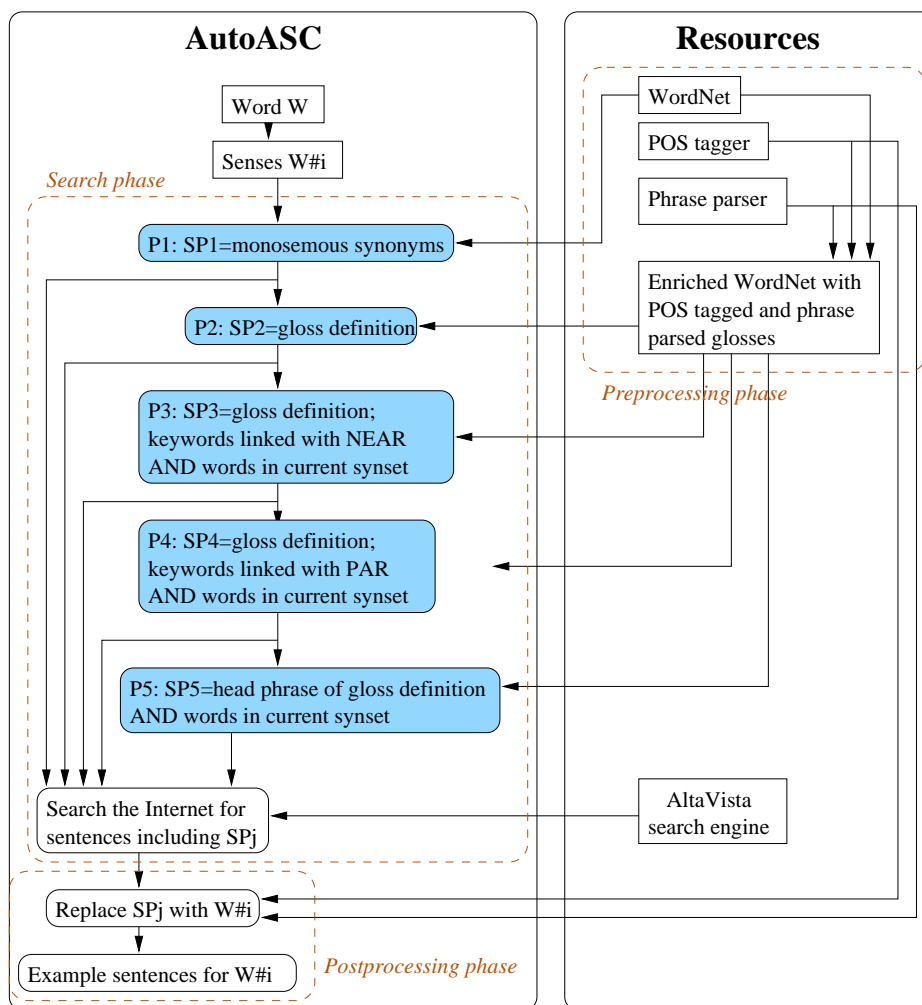


Fig. 1. The architecture of AutoASC.

- Step 2.* Eliminate the explanatory part of the definition, such as words included in brackets, or phrases starting with *as of*, *of*, *as in*, *as for*, etc.
- Step 3.* Part of speech tag the definition using Brill's tagger.²
- Step 4.* If the definition includes several phrases or sentences separated by semicolon, then each of these phrases is considered as an independent definition.
- Step 5.* Syntactically parse the definitions, i.e. detect the noun phrases, verb phrases, preposition attachments.¹⁵
- Step 6.* Based on the parsing from the previous step and the position of the *or* conjunction, create definitions with maximum one verb phrase and one noun phrase. For example, the definition for **better#1** 'to make better in quality or more valuable' will be separated into two definitions 'to make better in quality' and 'to make more valuable'.

2.2. Search

During the search phase, we have to identify a set of search phrases, each composed by one or several words, identical in meaning with the original word $W\#i$ and having the property that they uniquely identify the sense $\#i$ of the word W . Such a search phrase is then used to locate sentences on the Internet; these sentences will become, after the postprocessing phase, possible valid examples for $W\#i$. The search phrases are determined using the procedures below.

PROCEDURE P1. Determine a monosemous synonym, from the $W\#i$ synset. If such a synonym exists, this will constitute the search phrase.

Rationale. The context of a word is determined by the sense of that word. In the case of monosemous words, the context does not depend anymore on the sense of the word and is determined only by the word as a lexical string.

We performed several tests by considering also the direct hyponyms and direct hypernyms as possible relatives; the sentences we gathered using such words proved to give less representative examples than using the definition from the glosses (procedure P2.). Based on these empirical observations, we restricted the patterns for procedure P1 to synonymy relations.

Example. The noun *remember#1* has *recollect* as a monosemous synonym. Thus the search phrase for this word is *recollect*.

PROCEDURE P2. Parse the gloss, as explained above in this section. After the parse phase, we are left with a set of definitions, each of them constituting a search phrase.

Rationale. The role of a dictionary is to give definitions which uniquely identify the meaning of the words. Thus, the definition is specific enough to determine the context in which a particular word could appear.

Example. The verb *produce#5* has the definition (**bring onto the market or release, as of an intellectual creation**). The search phrase depicted from the definition is *bring onto the market* (the other possible definition *release* is eliminated, as being an ambiguous word).

PROCEDURE P3. Parse the gloss. Keep only the noun and verb phrases and concatenate them with the NEAR search-operator. The query is strengthened by adding the words from the current synset, using the AND search-operator.

Rationale. Using a query formed with the NEAR operator increases the number of hits but reduces the precision of the search; for this, we reinforce the query with words from the synset. This is based on the idea of one sense per discourse, as presented in Ref. 5.

Example. The synset of *produce#6* is {**grow, raise, farm, produce**} and it has the definition (**cultivate by growing**). This results in the following search phrase: *cultivate NEAR growing AND (grow OR raise OR farm OR produce)*.

PROCEDURE P4. Parse the gloss. Keep only the noun and verb phrases and concatenate them with the PAR (or paragraph) search-operator. This new lexical operator has been defined in Refs. 12 and 13: the condition imposed by PAR is that the words it connects occur in the same paragraph. This new operator can be regarded as a more relaxed NEAR. Again, the query is strengthened by adding the words from the current synset, using the AND search-operator.

Rationale. If the search phrase formed with the previous procedure, using the NEAR operator, leads to no results or only a few results, we need to relax even more the constraints of the query. The AND operator is too weak, thus it is necessary to have a new connector which fills the gap between AND and NEAR, and the PAR lexical operator does just that.

Example. The synset of *table#1* is {**table, tabular array**} and it has the definition (**a set of data arranged in rows and columns**). Using the PAR operator, we obtain the following search phrase: *set of data PAR rows PAR columns AND (table OR tabular array)*. This means that the system will locate those documents in which *set of data, rows* and *columns* occur in the same paragraph, and one of the words *table* or *tabular array* are also found in the document.

PROCEDURE P5. Parse the gloss. Keep only the head phrase combined with the words from the synset using the AND operator, as in procedure P3.

Rationale. If the search phrase determined during the previous procedure does not give any hits, the query can be relaxed even more by keeping only the head phrase. Again, a reinforcement is achieved by appending to the query the words from the synset.

Example. The synset of *company#5* is {**party, company**}, and the definition is (**band of people associated temporarily in some activity**). The search phrase for this noun becomes: *band of people AND (party OR company)*.

The search on the Internet with the queries formed with procedures P1–P5 results in several documents. A maximum of 1000 documents can be acquired for each search phrase, due to a limitation imposed by AltaVista that allows only the first 1000 hits resulting from a search to be accessed. From these texts, only the sentences containing the search phrases are extracted.

2.3. Postprocessing

The examples gathered during the search phase contain search phrases, which should have the same part of speech functionality as the original word *W*. If the search phrase consists of a single word, then part of speech tagging² is enough to check if it has the same functionality as the original word *W*. If the search phrase consists of several words, then further syntactic parsing is needed to determine if it is a noun, verb, adjective or adverb phrase and whether or not it has the same functionality as *W*. Those examples containing the search phrase with a different part of speech/syntactic tag with respect to the original word are eliminated. In

the remaining collection of examples, the search phrase is replaced with the original word, labeled with the appropriate sense number, i.e. $W\#i$.

3. AN EXAMPLE

As an example, consider the acquisition of sense tagged corpora for the noun *plane*. As defined in WordNet 1.6, *plane* is a common word with a polysemy count of 5. The synsets and the associated glosses for each of the senses of *plane* are presented in Fig. 2.

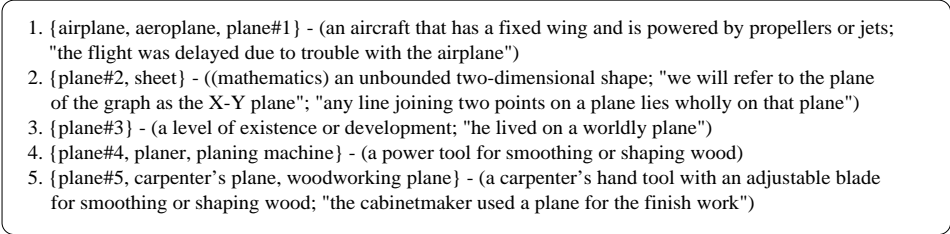
- 
1. {airplane, aeroplane, plane#1} - (an aircraft that has a fixed wing and is powered by propellers or jets; "the flight was delayed due to trouble with the airplane")
 2. {plane#2, sheet} - ((mathematics) an unbounded two-dimensional shape; "we will refer to the plane of the graph as the X-Y plane"; "any line joining two points on a plane lies wholly on that plane")
 3. {plane#3} - (a level of existence or development; "he lived on a worldly plane")
 4. {plane#4, planer, planing machine} - (a power tool for smoothing or shaping wood)
 5. {plane#5, carpenter's plane, woodworking plane} - (a carpenter's hand tool with an adjustable blade for smoothing or shaping wood; "the cabinetmaker used a plane for the finish work")

Fig. 2. Synsets and associated glosses for the different senses of the noun *plane*.

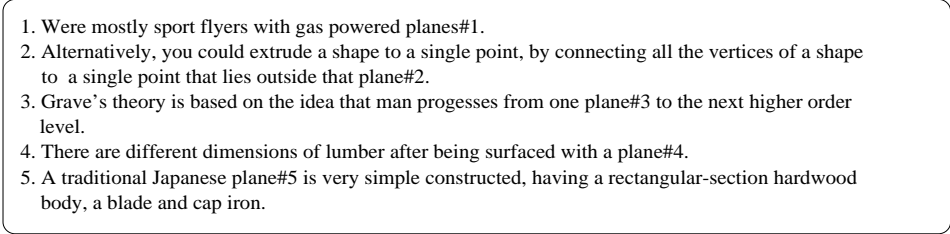
- 
1. Were mostly sport flyers with gas powered planes#1.
 2. Alternatively, you could extrude a shape to a single point, by connecting all the vertices of a shape to a single point that lies outside that plane#2.
 3. Grave's theory is based on the idea that man progresses from one plane#3 to the next higher order level.
 4. There are different dimensions of lumber after being surfaced with a plane#4.
 5. A traditional Japanese plane#5 is very simple constructed, having a rectangular-section hardwood body, a blade and cap iron.

Fig. 3. Context examples for various senses of the noun *plane*.

For each of the senses of the noun *plane*, Table 1 presents the search phrases (denoted by SP in this table) created by applying procedures $P1-P5$, and the number of sentences found with each of these search phrases.

Out of all the sentences extracted using the AltaVista search engine, we keep a maximum of 50 sentences for each sense of the noun *plane*, using the search phrases from Table 1 in ascending order of preference. In Fig. 3, an example sentence for each sense of *plane* is presented.

Table 2 summarizes the results: for each possible meaning of the noun *plane*, we present the total number of sentences found by our system, the number of examples manually checked, as well as the number of occurrences found in the SemCor files. Out of 209 sentences manually checked, 207 have been considered correct based on human judgment, thus a similarity of 99% with respect to manually tagged data. The total number of examples found by AutoASC is 7010, as opposed to only 39 examples which could be extracted from SemCor.

Table 1. Search phrases and number of examples found for each sense of the noun *plane*.

Sense #	Procedure	Search phrase	Examples
plane#1	<i>P1</i>	SP = airplane; SP = aeroplane	2000
	<i>P2</i>	SP = an aircraft that has fixed a wing and is powered by propellers or jets	0
	<i>P3</i>	SP = aircraft NEAR wing NEAR propellers NEAR jets AND (plane OR airplane OR aeroplane)	4
	<i>P4</i>	SP = aircraft PAR wing PAR propellers PAR jets AND (plane OR airplane OR aeroplane)	41
	<i>P5</i>	SP = aircraft AND (plane OR airplane OR aeroplane)	1000
plane#2	<i>P1</i>	–	0
	<i>P2</i>	SP = unbounded two-dimensional shape	0
	<i>P3</i>	SP = unbounded NEAR two dimensional NEAR shape AND (plane OR sheet)	1
	<i>P4</i>	SP = unbounded PAR two dimensional PAR shape AND (plane OR sheet)	1
	<i>P5</i>	SP = two-dimensional shape	8
plane#3	<i>P1</i>	–	0
	<i>P2</i>	SP = level of existence; SP = level of development	2000
	<i>P3</i>	–	0
	<i>P4</i>	–	0
	<i>P5</i>	–	0
plane#4	<i>P1</i>	SP = planer; SP = planing machine	1218
	<i>P2</i>	SP = power tool for smoothing or shaping wood	0
	<i>P3</i>	SP = power tool NEAR smoothing NEAR shaping NEAR wood AND (plane OR planer OR planing machine)	0
	<i>P4</i>	SP = power tool PAR smoothing PAR shaping PAR wood AND (plane OR planer OR planing machine)	0
	<i>P5</i>	SP = power tool AND (plane OR planer OR planing machine)	313
plane#5	<i>P1</i>	SP = carpenter's plane; SP = woodworking plane	87
	<i>P2</i>	SP = carpenter's hand tool with an adjustable blade for smoothing or shaping wood	0
	<i>P3</i>	SP = carpenter NEAR hand tool NEAR blade NEAR smoothing NEAR shaping NEAR wood AND (plane OR carpenter's plane OR woodworking plane)	0
	<i>P4</i>	SP = carpenter PAR hand tool PAR blade PAR smoothing PAR shaping PAR wood AND (plane OR carpenter's plane OR woodworking plane)	0
	<i>P5</i>	SP = hand tool AND (plane OR carpenter's plane OR woodworking plane)	337

Table 2. Results obtained for the five senses of the noun *plane*.

Word	Examples in SemCor	Total # examples acquired	Examples manually checked	Correct examples
plane#1	21	3045	50	50
plane#2	16	10	9	8
plane#3	2	2000	50	50
plane#4	0	1531	50	49
plane#5	0	424	50	50
TOTAL	39	7010	209	207

4. RESULTS

The system was tested on a set of ten words, randomly selected from a set of words with a polysemy count greater or equal to three, as defined in WordNet. The set consists of five nouns: *chair*, *problem*, *acquisition*, *plane*, *table* and five verbs: *clarify*, *drink*, *speak*, *indicate* and *believe*. This led to a set of 46 different word meanings. For each of these words, the method presented here was applied and example contexts were acquired. We extracted a maximum of 50 examples from the Internet, for each of these 46 concepts. The documents obtained from a search performed with the AltaVista search engine on the Internet are ranked based on their relevance to the input search phrase. We checked these documents in order, starting with the top ranked ones, such as to collect a set of maximum 50 sentences containing the search phrase. We then manually checked these sentences for sense tagging correctness.

Table 3 presents the polysemy for each of the words, the total number of occurrences within SemCor (*brown1*, *brown2* and *brownv* semantic concordances) and the total number of examples acquired using each of the procedures *P1–P5*. The numbers marked with a star indicate that the corresponding procedure could not be applied for some or all the senses of the word. This happens in the case of words with short definitions, as for example the definition of *drink#1*: (`consume alcohol`). Applying procedures *P2* through *P5* results in the same search phrase, identical with the definition; thus, the number of examples found on the Internet for this search phrase is counted only as a result for *P2*, and the number of examples gathered with the other procedures *P3–P5* are marked with a star.

In Table 4, we present the total number of examples manually checked (the columns denoted with **a**) and the number of examples considered to be correct, based on human judgment (the columns denoted with **b**) for each of the five procedures *P1–P5*. For 46 different meanings considered, a total of 2071 examples have been automatically acquired and then manually checked. Out of these 2071 examples, 1796 proved to be correct based on human judgment, yielding an accuracy of 87% such that the tag assigned with our method was the same as the tag assigned by human judgment.

Table 3. Results obtained for example contexts gathered for ten words.

Word	Polysemy count	Examples in SemCor	Examples acquired with procedure					Total examples acquired
			P_1	P_2	P_3	P_4	P_5	
chair	4	37	5087	712	1*	1*	53*	5801
problem	3	199	0	665	19	92	1009	1785
acquisition	4	11	1000	90	3*	20*	644*	1757
plane	5	39	3305	2000	5*	42*	1658*	7010
table	6	81	64	270	1555	2108	5587	9584
clarify	4	7	1000	4557	0*	0*	0*	5557
drink	5	28	1030	2970	0*	0*	0*	4000
speak	5	147	1000	3184	25*	25*	1000*	5234
indicate	5	183	0	2761	0*	2*	1000*	3763
believe	5	215	0	2821	112*	59*	1000*	3992
TOTAL	46	947	12486	20030	1720	2349	11951	48536

Table 4. Results obtained for example contexts gathered for ten words.

Word	Examples manually checked(a) and Correct examples (b) for procedure										Total examples manually checked	Correct examples
	P_1		P_2		P_3		P_4		P_5			
	a	b	a	b	a	b	a	b	a	b		
chair	150	150	0	0	1	1	1	1	3	3	155	155
problem	0	0	51	51	19	15	30	25	9	8	109	99
acquisition	50	50	50	45	0	0	17	10	83	50	200	155
plane	150	149	50	50	1	1	0	0	8	7	209	207
table	52	52	50	15	5	5	20	17	173	158	300	247
clarify	50	50	102	102	0	0	0	0	0	0	152	152
drink	80	69	116	100	0	0	0	0	0	0	196	169
speak	50	50	152	145	25	0	23	0	0	0	250	195
indicate	0	0	200	194	0	0	2	0	48	10	250	204
believe	0	0	153	128	75	66	22	19	0	0	250	213
TOTAL	582	570	924	830	126	88	115	72	324	236	2071	1796

Using this method, very large corpora can be generated. As reported in Table 3, for a total of 10 words, 48,536 examples were acquired, 50 times more than the 947 examples found in SemCor for these words. Even though this corpora is noisy, it is much easier and less time consuming to check an already existing tagged corpora for correctness, than to start tagging free text from scratch.

Procedures evaluation To evaluate the performance of the procedures, we define three factors:

1. *full-recall*, defined as the number of examples gathered with procedure P_i over the total number of examples acquired (here: 48,536).

2. *recall*, defined as the number of examples gathered with procedure P_i over the total number of examples manually checked (here: maximum 50 for each word sense).
3. *precision*, defined as the number of examples found correct with procedure P_i over the number of examples manually checked for the same procedure P_i

Table 5 presents the value of these factors for each of the procedures $P1$ – $P5$. Procedure $P1$ is the easiest to apply and is straightforward: sentences including a monosemous synonym of a given word become valid examples for that word. But this procedure, as suggested by the table above, found only about 25% of the cases. Procedure $P2$, which makes use of the whole definition of a given word as found in WordNet, accounts for the majority of the results: about 41% of the examples for a given concept are gathered with procedure $P2$. As expected, procedure $P1$ has the highest precision: in 97% of the cases, the examples found with $P1$ proved to be correct. Still, the highest value for precision plus recall is achieved with procedure $P2$; this shows how valuable is the usage of word definitions in finding valid examples for word senses.

Table 5. *Full-recall*, *recall* and *precision* for procedures $P1$ – $P5$.

Factor	Procedure				
	$P1$	$P2$	$P3$	$P4$	$P5$
<i>full-recall</i>	25.7%	41.2%	3.5%	4.8%	24.8%
<i>recall</i>	28.1%	44.6%	6.1%	5.5%	15.7%
<i>precision</i>	97.9%	89.8%	69.8%	62.6%	72.8%

An important observation related to the number of examples which can be obtained is that this number does not always correlate with the frequency of senses, thus classifiers using such a corpora will have to establish prior probabilities.

5. PREVIOUS WORK

Several approaches have been proposed in previous research for the automatic acquisition of sense tagged corpora.

In Ref. 5, a bilingual French–English corpus is used. For an English word, the classification of contexts in which various senses of that word appear is done based on the different translations in French for the different word meanings. The problem with this approach is that aligned bilingual corpora are very rare; also, different senses of many polysemous words in English often translate to the same word in French, for such words to acquire examples with this method is impossible.

Another approach for creating training and testing materials is presented in Ref. 16. Here, the Roget’s categories are used to collect sentences from a corpus. For example, for the noun *crane* which appears in both Roget’s categories *animal* and *tool*, he uses words in each category to extract contexts from *Grolier’s Encyclopedia*.

In Ref. 17, Yarowski proposed the automatic augmentation of a small set of seed collocations to a larger set of training materials. He located examples containing the seeds in the corpus and analyzed these to find new patterns; then, he retrieved examples containing these patterns. WordNet is suggested here as a source for seed collocations. In contrast, our method tries to find example sentences including words or expressions similar in meaning with a particular sense of the input word, but which have the property that uniquely identifies that sense of the word. On the other hand, the algorithm presented in Ref. 17 iteratively builds a set of words which are likely to appear in the context of a given sense of the input word. This set of context words is used to identify example sentences for the different meanings of the input word. The accuracy obtained is about 95% when salient words of dictionary definitions are used as seeds. This precision is higher than the one achieved with AutoASC (87%); but the advantage of our system is that no human intervention is needed in the process of acquiring sense tagged corpora.

In Ref. 8, a method based on the monosemous words from WordNet is presented. For a given word, its monosemous lexical relatives provide a key for finding relevant training sentences in a corpus. An example given in their paper is the noun *suit* which is a polysemous word, but one sense of it has *business suit* as monosemous hyponym, and another has *legal proceeding* as a hypernym. By collecting examples containing *business suit* and *legal proceeding*, two sets of contexts for the senses of *suit* are built. Even though this method gives high accuracy results for WSD (about 85% accuracy) respect to manually tagged materials, its applicability for a particular word *W* is limited by the existence of monosemous “relatives” (i.e. words semantically related to the word *W*) for the different senses of *W* and by the number of appearances of these monosemous “relatives” in the corpora. Restricting the semantic relations to synonyms, direct hyponyms and direct hypernyms, they found that about 64% of the words in WordNet have monosemous “relatives” in the 30-million-word corpus of the *San Jose Mercury News*. Other tests performed on a set of 1100 words showed that only about 25% of the word senses of the polysemous words have monosemous synonyms.

Our approach tries to overcome these limitations in two ways: (1) by using other useful information that can be found in WordNet for a particular word, i.e. the word definitions provided in the glosses and (2) by using a very large corpus, formed by the texts electronically stored on the Internet.

6. CONCLUSION AND FURTHER WORK

In this paper we have presented AutoASC, a system that enables the acquisition of sense tagged corpora based on the information found in WordNet and on the very large collection of texts which can be found on the World Wide Web. An explanation uniquely identifying a word is provided either by its monosemous synonyms or by its definition. Several procedures are applied to determine such an explanation which will further constitute a search phrase. Based on this, several examples are automatically acquired from the World Wide Web, using an existing search engine.

The system has been tested on a set of 46 concepts and 2071 context examples for these words have been acquired. An accuracy of 87% was observed.

The majority of examples acquired with this system have been determined using procedures 2 through 5, which are based on the definitions found within the glosses in WordNet. This suggests the value of using word definitions for locating valid context examples for word senses.

As stated in Sec. 4, one of the main drawbacks of our system is the fact that the number of examples acquired for a word sense does not always correlate with the frequency of word senses in the English language. A solution to this problem would be to use sense frequencies derived from an existing sense tagged corpus (SemCor, for example). These frequencies can be used to normalize the number of examples extracted with AutoASC, such that these numbers will correlate with the natural frequency of word senses.

Further work will include the use of this method for automatic acquisition of very large corpora which will be used to test word sense disambiguation accuracy.

REFERENCES

1. Altavista, 1999, Digital Equipment Corporation, <http://www.altavista.com>
2. E. Brill, "A simple rule-based part of speech tagger," *Proc. 3rd Conf. Applied Natural Language Processing*, Trento, Italy, 1992.
3. R. Bruce and J. Wiebe, "Word sense disambiguation using decomposable models," *Proc. 32nd Annual Meeting of the Association for Computational Linguistics (ACL-94)*, LasCruces, NM, June 1994, pp. 139–146.
4. C. Fellbaum, *WordNet, An Electronic Lexical Database*, The MIT Press, 1998.
5. W. Gale, K. Church and D. Yarowsky, "One sense per discourse," *Proc. 4th DARPA Speech and Natural Language Workshop*, Harriman, New York, 1992, pp. 233–237.
6. S. Harabagiu and D. Moldovan, "Enriching the WordNet taxonomy with contextual knowledge acquired from text," *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, eds. S. Shapiro and L. Iwanska, AAAI/MIT Press, 1999.
7. N. Ide and J. Veronis, "Introduction to the special issue of word sense disambiguation: the state of the art," *Computational Linguistics* **24**, 21 (1998) 1–40.
8. C. Leacock, M. Chodorow and G. Miller, "Using corpus statistics and WordNet relations for sense identification," *Computational Linguistics* **24**, 21 (1998) 147–165.
9. R. Mihalcea and D. Moldovan, "Automatic acquisition of sense tagged corpora," *Proc. FLAIRS-99*, Orlando, FL, May 1999, pp. 293–298.
10. G. Miller, "WordNet: a lexical database," *Commun. ACM* **38**, 11 (1995) 39–41.
11. G. Miller, M. Chodorow, S. Landes, C. Leacock and R. Thomas, "Using a semantic concordance for sense identification," *Proc. 4th ARPA Human Language Technology Workshop*, 1994, pp. 240–243.
12. D. Moldovan and R. Mihalcea, "An WordNet-based interface to internet search engines," *Proc. FLAIRS-98*, Sanibel Island, FL, May 1998, pp. 275–280.
13. D. Moldovan and R. Mihalcea, "Improving the search on the internet by using WordNet and new lexical operators," *Internet Comput.* (1999), to appear.
14. H. Ng and H. Lee, "Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach," *Proc. 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*, Santa Cruz, 1996, pp. 40–47.

15. B. Srinivas, "Performance evaluation of supertagging for partial parsing," *Proc. 5th Int. Workshop on Parsing Technology*, Boston, September 1997.
 16. D. Yarowsky, "Word-sense disambiguation using statistical models of Roget's categories trained on large corpora," *Proc. COLING-ACL 1992*, Nantes, France, 1992, pp. 454–460.
 17. D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," *Proc. 33rd Annual Meeting of the Association for Computational Linguistics (ACL-95)*, Cambridge, MA, 1995, pp. 189–196.
-



Rada Mihalcea is a Ph.D. student in the Department of Computer Science at Southern Methodist University.

Her research interests are in the field of Natural Language Processing, particularly in word sense disambiguation, information extraction and question answering systems. She is a member of AAAI, ACL.



Dan Moldovan is a Professor in the Department of Computer Science and Engineering at Southern Methodist University. He earned a Ph.D. degree in electrical engineering and computer science from Columbia University in

1978.

His current research interests are in natural language processing, particularly in linguistic knowledge bases, text inference and question answering systems.