Instance Based Learning with Automatic Feature Selection Applied to Word Sense Disambiguation

Rada MIHALCEA
Department of Computer Science
University of North Texas
Denton, TX, 76203-1366
rada@cs.unt.edu

Abstract

We describe an algorithm for Word Sense Disambiguation (WSD) that relies on a lazy learner improved with automatic feature selection. The algorithm was implemented in a system that achieves excellent performance on the set of data released during the SENSEVAL-2 competition. We present the results obtained and discuss the performance of various features in the context of supervised learning algorithms for WSD.

1 Introduction

The task of Word Sense Disambiguation consists in assigning the most appropriate meaning to a polysemous word within a given context. A large range of applications, including machine translation, knowledge acquisition, information retrieval, information extraction, and others, require knowledge about word meanings, and therefore WSD algorithms represent a necessary step in all these applications. Starting with SENSEVAL-1 in 1999, WSD has received growing attention from the Natural Language Processing community, and motivates a continuously increasing number of researchers to develop WSD systems and devote time for finding solutions to this challenging problem.

The SENSEVAL[1] competitions provided a good environment for the development of supervised WSD systems, making freely available large amounts of sense tagged data. During SENSEVAL-1 in 1999, data for 35 words was made available adding up to about 20,000 examples tagged with respect to the Hector dictionary. The size of the tagged corpus increased with SENSEVAL-2 in 2001, when 13,000 additional examples were released for 73 polysemous words. This time, the semantic annotations were performed with respect to WordNet. The experiments and results reported in this paper pertain to the SENSEVAL-2 data. However, similar experiments were performed on the SENSEVAL-1 data with comparable results.

Most of the efforts in the WSD field were concentrated so far towards *supervised* learning algorithms, and these are the methods that usually achieve the best performance at the cost of low recall. Each sense tagged occurrence of a particular word is transformed into a feature vector, suitable for an automatic learning process. Two main decisions need to be made in the design of such a system: the set of features to be used and the learning algorithm. Commonly used features include surrounding words and their part of speech(Bruce and Wiebe, 1999), context keywords (Ng and Lee, 1996) or context bigrams (Pedersen, 2001), various syntactic properties (Fellbaum et al., 2001) etc. As for the learning methodology, a large range of algorithms have been employed, including neural networks (Leacock et al., 1998), decision trees (Pedersen, 2001), decision lists (Yarowsky, 2000), memory based learning (Veenstra et al., 2000) and others. An experimental comparison of seven learning algorithms used to disambiguate the meaning of the word *line* is presented in (Mooney, 1996).

We investigate in this paper the use of a lazy learner, namely instance based learning, to solve the semantic ambiguity of words in context. The main advantage of instance based learners is the fact that they consider every single training example when making a classification decision. This characteristic proves particularly useful for NLP problems, where training data is usually expensive and exceptions are important. On the other side, lazy learners, including

[1] http://www.itri.bton.ac.uk/events/senseval/

instance based learners, have the disadvantage of being easily misled by irrelevant features. In the algorithm described in this paper, this drawback is solved by improving the learner with a scheme for automatic feature selection.

The methodology presented here is integral part of a larger system that has the capability of performing both supervised and open-text WSD (Mihalcea, 2002). For reasons of clarity and space, we focus in this paper only on the description of the supervised component.

To our knowledge, instance based learning with per word automatic feature selection is a new approach in the WSD field, and we show that it leads to very good results. Previous work has considered the application of instance based learning with automatic feature selection for the problem of pronoun resolution (Cardie, 1996). In WSD, the work that is closest to ours was reported by (Bruce and Wiebe, 1999), where decomposable probabilistic models are used in combination with eager Naive Bayes algorithms.

2 Learning with automatic feature selection

Learning mechanisms for disambiguating word sense have a long tradition in the WSD field, including a large range of algorithms and feature types. For our system, we have decided for an instance based algorithm with information gain feature weighting. The reasons for this decision are threefold. First, it has been advocated that forgetting exceptions is harmful in language learning applications (Daelemans et al., 1999), and instance based algorithms are known for their property of taking into consideration every single training example when making a classification decision. Second, this type of algorithms have been successfully used in WSD applications (Veenstra et al., 2000). Finally, the last reason for our decision was the running time efficiency of these algorithms. We have initially used the MLC++[2] implementation, and later on switched to TiMBL[3] (Daelemans et al., 2001).

[2]Machine Learning library available at http://www.sgi.com/tech/mlc/docs.html

[3]Available from http://ilk.kub.nl/software.html. All TiMBL runs were made with the default settings, namely IB1 algorithm with gain ratio feature weighting, k-NN classification, no modified value difference metric (MVDM).

The main disadvantage of lazy learners, including instance based learning algorithms, is their high sensitivity to irrelevant features. Severe degradation in accuracy may result as a consequence of too many such features in the training examples. It turns out that a critical factor influencing the performance of an instance based learner is the selection of features employed during the learning process.

Our intuition was that different sets of features have different effects depending on the ambiguous word considered. Rather than creating a general learning model for all polysemous words, a separate feature space is built for each individual word. Usually, features are weighted using weighting schemes that are based on information gain, gain ratio, chi-squared or other information content measures. Feature weighting was clearly proven to be an advantageous approach for a large range of applications, including WSD. Still, weights are computed independently for each feature and therefore this strategy does not always guarantee to provide the best results. Sometimes it is better to leave features out than assign them even a small weight. We therefore face the problem of defining a procedure for feature selection that would ideally minimize the disambiguation error.

Variable sets of features have been successfully used in other Artificial intelligence applications. (Cardie, 1996) proposes a linguistic and cognitive biased approach for relative pronoun resolution. In (Aha and Bankert, 1994), features are selected using searching algorithms, with increased performance obtained in the problem of cloud types classification. (Domingos, 1997) introduces an algorithm for context sensitive feature selection, with different features selected for each instance in the training set. Various efficient search algorithms for the detection of optimal feature subsets are proposed in (Moore and Lee, 1994) with successful experiments performed on several synthetic datasets.

In our algorithm, features are automatically selected using a forward search algorithm. The classic approach is to build word experts via a learning process that determines the values for a pre-selected set of features. Instead, we first learn the set of features that would best model the word characteristics, and therefore we are

exploiting at maximum the idiosyncratic nature of words. It is only at a second stage that we actually build the word experts by determining the values for the set of features previously determined.

With this approach, we combine the advantages of instance based learning mechanisms that have the nice property of *"not forgetting exceptions"*, with an optimized feature selection scheme.

3 Main Algorithm

The corpus provided for each ambiguous word is first run through a preprocessing stage, where the text is annotated with lexical tags. Next, each example is transformed into a feature vector. Features are selected from a pool of features using an automatic selection algorithm. The train and test instances will therefore include only the features in the subset determined to be optimal by the selection algorithm.

Notice that training and testing corpora are extracted for each ambiguous word. This means that examples pertaining to the compound *"dress down"* are separated from the examples for the single word *"dress"*.

3.1 Preprocessing

During the preprocessing stage, SGML tags are eliminated, the text is tokenized, part of speech tags are assigned using Brill tagger (Brill, 1995), and Named Entities (NE) are identified with an *in-house* implementation of an NE recognizer. To identify collocations, we determine sequences of words that form compound concepts defined in WordNet (Miller, 1995). There are two possible problems with this approach. The first one concerns subsuming concepts, as in *"United States"* and *"United States of America"*. In such cases, priority is given to the longest sequence of words. The second possible conflict regards overlapping concepts, like the two different compounds *"English Channel"* and *"Channel Tunnel"* found in the text *"English Channel Tunnel"*. Here, we break the tie by keeping the last encountered collocation, with the only reason for this decision being the ease of implementation.

3.2 Algorithm for Automatic Feature Selection

The algorithm for automatic feature selection is sketched below.

---

function *AutomaticFeatureSelection*
- *generate a pool of features* $PF = \{F_i\}$
- *initialize the set of selected features with the empty set* $SF = \{\emptyset\}$
- *extract training and testing corpora for the given target ambiguous word.*
- **loop: for** *each feature* $F_i \in PF$
  - *run a 10-fold cross validation on the training set; each example in the training set contains the features in SF* **and** *the feature* F_i.
  - *determine the feature* F_i *leading to the best accuracy*
  - *remove* F_i *from PF and add it to SF*
- **goto loop** *until no improvements are obtained*

---

4 Features that are good indicators of word sense

There are several features acknowledged as good indicators of word sense, including surrounding words, part of speech tags, collocations, syntactic roles, keywords in contexts. More recently, other possible features have been investigated: bigrams in context, named entities, semantic relations with other words in context, etc.

We distinguish three types of features:

1. *0-param* features, which may be included in the optimal subset or not, without any parameters to set. For instance, the part of speech of a surrounding word is a *zero parameters* feature, since any learning example can either contain or omit this feature, without having to indicate a specific value.

2. *1-param* features, which, once selected, have one variable parameter that can be set to a specific value (alternatively, this parameter is left with its default value). As an example, consider the *context* feature (CF), which includes the words in a surrounding window of length K. Deciding the value for K implicitly means setting *one parameter* for this feature.

3. *2-param* features with two parameters associated. For example, one can select MX keywords representative for the context of

an ambiguous word, where a keyword is defined as a word that occurs at least MN times. Therefore, *two parameters* have to be set for this feature, MX and MN.

All features that have been considered so far are presented below. They form the *pool of features* PF from which features are selected using the algorithm described in Section 3.2. In the following, the ambiguous word is denoted with *AW*.

CW *Current word (0-param) The word AW itself. Notation: CW*

CP *Current part of speech (0-param) The part of speech of the word AW. Notation: CP*

CF *Contextual features (1-param) The words and parts of speech of K words surrounding AW. Notation: CF[=K], default=3*

COL *Collocations (1-param) Collocations (Ng and Lee, 1996) formed with maximum K words surrounding AW. Notation: COL[=K], default=3*

HNP *Head of noun phrase (0-param) The head of the noun phrase to which AW belongs, if any. Notation: HNP*

SK *Sense specific keywords (2-param) Maximum MX keywords occurring at least MN times (Ng and Lee, 1996) are determined for each sense of the ambiguous word. The value of this feature is either 0 or 1, depending if the current example contains one of the determined keywords or not. Notation: SK[=MN,MX], default=5,5*

B *Bigrams (2-param) Maximum MX bigrams occurring at least MN times are determined for all training examples. The value of this feature is either 0 or 1, depending if the current example contains one of the determined bigrams or not. Bigrams are ordered using the Dice coefficient(Pedersen, 2001). Notation: B[=MN,MX], default=5,20*

VB *Verb before (0-param) The first verb found before AW. Notation: VB*

VA *Verb after (0-param) The first verb found after AW. Notation: VA*

NB *Noun before (0-param) The first noun found before AW. Notation: NB*

NA *Noun after (0-param) The first noun found after AW. Notation: NA*

NEB *Named Entity before (0-param) The first Named Entity found before AW. Notation: NEB*

NEA *Named Entity after (0-param) The first Named Entity found after AW. Notation: NEA*

PB *Preposition before (0-param) The first preposition found before AW. Notation: PB*

PA *Preposition after (0-param) The first preposition found after AW. Notation: PA*

PRB *Pronoun before (0-param) The first pronoun found before AW. Notation: PRB*

PRA *Pronoun after (0-param) The first pronoun found after AW. Notation: PRA*

DT *Determiner (0-param) The determiner, if any, found before AW. Notation: DT*

New features can be easily added to the pool, with no changes required in the main algorithm. The system was initially tested with the SENSEVAL-1 data, and additional features were considered at that time to help towards performance. We decided not to use them in the current experiments, mainly for time considerations, since parsing is a highly computational intensive task.

PPT *Parse path (1-param) Maximum K parse components found on the path to the top of the parse tree (sentence top). Notation: PPT[=K], default=10. For instance, the value of this feature for the word school, given a parse tree (S (NP (JJ big) (NN house))), is NN, NP, S.*

SPC *Same parse phrase components (1-param) Maximum K parse components found in the same phrase as AW. Notation: SPC[=K], default=3. For the example above, this feature would be set to JJ, NN.*

5 Results on SENSEVAL-2 Data

The overall performance of the system evaluated on the test words released during the SENSEVAL-2 *English lexical sample* task is 63.8% for fine-grained scoring (71.2% for coarse-grained scoring). These results are comparable with the best performing systems participating in the competition.

Table 1 presents the results obtained for the lexical sample task, for 73 ambiguous words, including 29 nouns, 15 adjectives and 29 verbs. For each word, the table shows: number of examples in the training and test sets; features automatically selected as a result of the algorithm in Section 3.2; 10-fold cross validation precision obtained on the training data with the selected features; the precision for fine-grained and coarse-grained scoring when all features in PF are considered (i.e. no per word feature selection is performed); finally, we show the precision for fine-grained and coarse-grained scoring computed on the test data when features are automatically selected on an individual word basis.

For the *1-param* and *2-param* features, there is a range of values allowed for their parameters. This range was empirically set to [1-5] for the *1-param* features, respectively [1-10] for the *2-param* features. It means that, for instance, CF can be set to CF=1, CF=2, CF=3, CF=4 or CF=5. The selection of the best value is per-

word.pos	Size train	Size test	Features	10-fold valid.	All features fine	All features coarse	Feature selection fine	Feature selection coarse
art.n	194	98	CF=1 HNP B=2,5 VB NB	60.6%	67.3%	69.4%	71.4%	74.5%
authority.n	183	92	CW CP COL=1 VB NB	62.2%	75.0%	91.4%	70.7%	91.3%
bar.n	264	151	CW CP CF=1 COL=1 B=5,3 VB NB NEA	60.8%	55.6	64.2%	62.3%	74.5%
bum.n	80	45	CW NA NEA	86.2%	80.0%	82.2%	77.8%	80.0%
chair.n	137	69	CW	92.3%	85.5%	87.0%	85.5%	88.4%
channel.n	138	73	CP NB	43.0%	49.3%	57.5%	46.6%	56.2%
child.n	129	64	CW CP CF=1 COL=1 B=5,3 NB NEB DT	76.1%	60.9%	60.9%	68.8%	68.8%
church.n	128	64	CW CP CF=2 COL=1 B=5,1	64.4%	54.7%	58.8%	56.2%	56.2%
circuit.n	169	85	CP CF=3 B=5,1 VB	51.6%	58.8%	64.7%	58.8%	62.4%
day.n	289	145	CP CF=2 HNP NEB PB	78.0%	69.7%	69.7%	76.1%	77.3%
detention.n	63	32	any	-	90.6%	90.6%	87.5%	87.5%
dyke.n	58	28	CW CF=2 SK=5,2	91.4%	85.7%	85.7%	89.3%	89.3%
facility.n	114	58	CP COL=1 VB PRB	74.5%	72.4%	100.0%	79.3%	98.3%
fatigue.n	76	43	CP B=5,3 NB	86.6%	81.4%	86.0%	88.4%	90.7%
feeling.n	102	51	CP CF=1 COL=3 HNP NEA	64.0%	66.7%	68.6%	74.5%	74.5%
grip.n	100	51	CP CF=3 COL=2 PB DT	60.0%	35.3%	56.9%	41.2%	58.8%
hearth.n	64	32	CP CF=1 HNP	66.7%	59.4%	81.2%	75.0%	87.5%
holiday.n	62	31	CP	96.0%	93.5%	96.8%	93.5%	96.8%
lady.n	103	53	CW HNP	84.0%	79.2%	96.2%	88.7%	94.3%
material.n	140	69	CW CP COL=1 B=2,5 VA NEA	53.3%	60.9%	68.1%	56.5%	60.9%
mouth.n	118	60	CP COL=1 VB NB PB	65.7%	58.3%	93.3%	65.0%	93.3%
nation.n	75	37	CP	80.0%	51.4%	51.4%	54.1%	54.1%
nature.n	92	46	CP DT	58.0%	76.1%	82.6%	69.6%	80.4%
post.n	150	79	CW CP CF=1 COL=2	74.6%	46.4%	49.3%	64.6%	68.4%
restraint.n	91	45	CP COL=2 HNP B=2,5 VB NB PA	67.3%	62.2%	73.3%	62.2%	71.1%
sense.n	107	53	CP CF=1 B=3,3 NEB PB	74.5%	71.7%	73.6%	75.5%	74.4%
spade.n	64	33	CP CF=1 COL=2	94.0%	78.8%	78.8%	97.0%	97.0%
stress.n	78	39	CP COL=2 B=5,2	68.0%	48.7%	82.1%	64.1%	89.7%
yew.n	57	28	CF=1	94.0%	82.1%	100.0%	89.3%	100.0%
TOTAL.N	3,523	1,759	-	-	65.6%	73.9%	69.5%	76.6%
blind.a	105	55	HNP	70.0%	74.5%	74.5%	85.5%	85.5%
colourless.a	68	35	CW CP CF=1 COL=1 SK=3,3	85.7%	54.3%	54.3%	48.6%	48.6%
cool.a	103	52	CF=1 COL=2 HNP VB PB PRB DT	56.1%	44.2%	44.2%	51.9%	51.9%
faithful.a	47	23	CW	68.0%	65.2%	65.2%	87.0%	87.0%
fine.a	139	70	CP CF=2 HNP B=5,1 NA	46.0%	51.4%	51.4%	54.3%	54.3%
fit.a	57	29	CF=1 B=3,3 VB NA	85.0%	79.3%	79.3%	82.8%	82.8%
free.a	165	82	CP CF=1 COL=2	65.0%	52.4%	52.4%	58.5%	58.5%
graceful.a	56	29	CW	87.0%	86.2%	86.2%	79.3%	79.3%
green.a	190	94	CP VA	80.0%	76.6%	76.6%	79.8%	79.8%
local.a	75	38	CP NA	88.0%	71.1%	71.1%	81.6%	81.6%
natural.a	205	103	CP CF=1 HNP VB NB NEB PRA	50.0%	52.4%	52.4%	56.3%	56.3%
oblique.a	56	29	CW CP CF=1 COL=4 B=3,3	84.0%	79.3%	79.3%	86.2%	86.2%
simple.a	130	66	CP CF=1 COL=2 HNP NA PB PRA DT	53.3%	40.9%	40.9%	53.0%	53.0%
solemn.a	52	25	CP COL=1 DT	92.8%	96.0%	96.0%	96.0%	96.0%
vital.a	74	38	CW CP NB	88.7%	92.1%	92.1%	94.7%	94.7%
TOTAL.A	1,535	768	-	-	63.4%	63.4%	68.8%	68.8%
begin.v	557	280	CF=1 NA	80.40%	86.8%	86.8%	87.5%	87.5%
call.v	132	66	CF=1 COL=2 VB NB DT	70.00%	40.9%	66.7%	40.9%	66.7%
carry.v	132	66	CW CP COL=1 NB	35.00%	42.4%	53.0%	39.4%	50.0%
collaborate.v	57	30	CW CP CF=1	95.80%	90.0%	90.0%	90.0%	90.0%
develop.v	133	69	CW CP B=2,5 NA PB	22.50%	31.9%	50.7%	36.2%	49.3%
draw.v	82	41	CF=2 COL=2 NEB	11.00%	24.4%	34.1%	31.7%	43.9%
dress.v	119	59	CP CF=1 NB NA PB	57.50%	50.8%	86.4%	57.6%	86.4%
drift.v	63	32	CW CP CF=2 COL=3 HNP NEB PA	22.00%	53.1%	53.1%	59.4%	62.5%
drive.v	84	42	CW CP CF=2 PRA DT	45.00%	52.4%	76.2%	52.4%	69.0%
face.v	186	93	CP	84.00%	75.3%	89.2%	81.7%	90.3%
ferret.v	2	1	any	-	100.0%	100.0%	100.0%	100.0%
find.v	132	68	CP CF=2 SK=5,2	10.00%	25.0%	36.8%	29.4%	39.7%
keep.v	133	67	CP B=3,3	38.00%	40.3%	40.3%	44.8%	46.3%
leave.v	132	66	CP CF=1 COL=3 NEA	28.90%	40.9%	45.5%	47.0%	53.0%
live.v	129	67	CP NA	63.00%	61.2%	61.2%	67.2%	68.7%
match.v	86	42	CW CP HNP SK=5,5 NA	26.40%	45.2%	64.3%	40.5%	59.5%
play.v	129	66	CW CP CF=4 COL=4 VB NA	21.00%	34.8%	37.9%	50.0%	51.5%
pull.v	122	60	CP COL=1 HNP B=2,10 SK=5,5	23.00%	56.7%	70.0%	48.3%	68.3%
replace.v	86	45	CP COL=3 SK=5,1 B=3,2	54.00%	42.2%	93.3%	44.4%	88.9%
see.v	131	69	CW CP CF=2 SK=4,4 PB	23.00%	33.3%	37.7%	37.7%	42.0%
serve.v	100	51	CP CF=4 HNP B=5,5 VA NEB PRB PRA	36.00%	56.9%	60.8%	49.0%	54.9%
strike.v	104	54	CW CP CF=2 NEB	23.00%	42.6%	48.1%	38.9%	51.9%
train.v	125	63	CW CP CF=2 COL=4 NA PB PA DT	34.00%	38.1%	55.6%	41.3%	52.4%
treat.v	88	44	CP CF=3 COL=2 VB NEA PA PRB PRA	36.00%	38.6%	56.8%	63.6%	79.5%
turn.v	131	67	CP CF=2 VB NA PA PRB	30.70%	31.3%	47.8%	35.8%	53.7%
use.v	147	76	CW CP NA VA PRB	65.00%	60.5%	84.2%	72.4%	84.2%
wander.v	100	50	CP PA	81.00%	66.0%	84.0%	74.0%	90.0%
wash.v	25	12	CW CP CF=2 COL=2 SK=3,5 NEA	32.00%	66.7%	75.0%	66.7%	83.3%
work.v	119	60	CW CP CF=2 COL=2 B=3,3 NA PA	42.00%	35.0%	50.0%	43.3%	58.3%
TOTAL.V	3,673	1,857	-	-	52.5%	64.3%	56.4%	67.0%

Table 1: Training and test sizes, optimal feature sets and precisions for (1) 10-fold cross validation on training data; fine-grained and coarse-grained on test data using (2) all features; (3) per word feature selection.

formed using the same algorithm.

As mentioned earlier, collocations are identified since the preprocessing stage and the learning process is applied separately on each word. Therefore, the compound *"call for"* has training and test data different from the verb *"call"*, and consequently features are selected in a distinct process. Due to space limitations, Table 1 shows

the features selected only for single words.

When no training data is provided (as it was the case with the SENSEVAL-2 verb *"keep going"*), the first sense is applied by default. Also, when the training set size is smaller than 15 examples, the automatic feature selection algorithm is not invoked, instead a default set of features is used (CW CP CF=1 COL=1).

5.1 Discussion

Table 2 lists the number of times each feature was used in the semantic disambiguation of nouns, verbs and adjectives. The most often used features turn out to be CW, CP, CF and COL, which are also the features most frequently mentioned in the literature. Almost all words took advantage of the current part of speech (CP) feature. This is in agreement with (Stevenson and Wilks, 2001), who have emphasize the major role played by part of speech in WSD. It is interesting to observe that in terms of words in context, bigrams seem to be more effective than simple keywords. Also, the best setting for the CF feature was found to be one or two words window.

	Part of speech			
	Noun	Verb	Adjective	Total
Words	29	29	15	73
Features				
CW	10	13	9	32
CP	22	25	14	61
CF	14	18	8	40
COL	13	12	6	31
HNP	6	4	5	15
SK	1	6	3	10
B	10	6	3	19
VB	7	4	3	14
VA	1	2	1	4
NB	8	3	2	13
NA	1	10	4	25
NEB	3	4	1	8
NEA	4	3	0	7
PB	4	4	2	10
PA	1	6	0	7
PRB	1	4	1	6
PRA	0	3	2	5
DT	3	3	3	9
Total	109	130	66	306

Table 2: Feature distribution for nouns, verbs, adjectives

In terms of average number of features, the semantic disambiguation of nouns requires the smallest number of features (3.7), followed by adjectives (4.4) and verbs (4.5). These statistics are not yet conclusive, since they are computed for a small number of words, but they are indicative for the complexity of the task for various parts of speech. Further investigations and larger amounts of data will eventually confirm this preliminary conclusion.

The overall performance of the system when the module for per word feature selection is disabled and all features in PF are employed is 59.8% (68.1%). The increase in error rate is therefore about 11% with respect to the case when per word feature selection is employed.

We also performed an experiment where the feature selection algorithm consists in finding features that perform best over all 73 words. The set of feature determined with this simplified approach is "CW CP CF=1 COL=1". The overall performance when this constant set of features is employed is 59.6% (67.4%). Again, the per word feature selection is proved to produce better results.

Additionally, there were several interesting cases encountered in the SENSEVAL data, justifying our approach of using automatic feature selection. The influence of a feature greatly depends on the target word: a feature can increase the precision for a word, while making things worse for another word. For instance, a word such as *free* does not benefit from the SK feature, whereas *colourless* gains almost 7% in precision when this feature is used.

free.a[CW CP CF=1 SK=3,3]	→	57.85%
free.a[CW CP CF=1]	→	63.57%
colorless.a[CW CP CF=1]	→	78.57%
colorless.a[CW CP CF=1 SK=3,3]	→	85.71%

Another interesting example is constituted by the noun *chair*, disambiguated with high precision by simply using the current word (CW) feature. This is explained by the fact that the most frequent senses are **C***hair* meaning *person* and **c***hair* meaning *furniture*, and therefore the distinction between lower and upper case spellings makes the distinction among the different meanings of this word.

The noun *detention* has the same precision computed during several 10-fold cross validation runs, independent on the feature or combination of features used. This is because one of its two senses occurs in 97% of the examples, and hence it statistically dominates the other sense.

There were several other interesting cases, including the adjective *local* with a 20% gain in

precision by simply using the feature NA, the word *faithful* best disambiguated with the CW feature, and others.

6 Conclusion

Instance based learning with automatic feature selection is a new approach in the WSD field. The algorithm was implemented in a system that achieves excellent performance on the data released during the SENSEVAL-2 *English lexical task*. The feature selection process is completely automated and it practically creates a classifier tailored to the behaviour of each specific word.

Acknowledgments
The author would like to thank the anonymous reviewers for their helpful suggestions and constructive comments, which helped improving the quality of this manuscript.

References

D.W. Aha and R.L. Bankert. 1994. Feature selection for case-based classification of cloud types: An empirical comparison. In *Proceedings of the AAAI'94 Workshop on CaseBased Reasoning*, pages 106–112, Seattle, WA.

E. Brill. 1995. Transformation-based error driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–566, December.

R. Bruce and J. Wiebe. 1999. Decomposable modeling in natural language processing. *Computational Linguistics*, 25(2):195–207.

C. Cardie. 1996. Automating feature set selection for case-based learning of linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP*, pages 113–126, Somerset, New Jersey.

W. Daelemans, A. van den Bosch, and J. Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1-3):11–34.

W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. 2001. Timbl: Tilburg memory based learner, version 4.0, reference guide. Technical report, University of Antwerp.

P. Domingos. 1997. Context-sensitive feature selection for lazy learners. *Artificial Intelligence Review*, (11):227–253.

C. Fellbaum, M. Palmer, H.T. Dang, L. Delfs, and S. Wolf. 2001. Manual and automatic semantic annotation with WordNet. In *WordNet and Other lexical resources: NAACL 2001 workshop*, pages 3–10, Pittsburgh.

C. Leacock, M. Chodorow, and G.A. Miller. 1998. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics*, 24(1):147–165.

R. Mihalcea. 2002. Word sense disambiguation using pattern learning and automatic feature selection. *Journal of Natural Language Engineering (to appear)*.

G. Miller. 1995. Wordnet: A lexical database. *Communication of the ACM*, 38(11):39–41.

R. Mooney. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proceedings of the 1996 Conference on Empirical Methods in Natural Language Processing (EMNLP-1996)*, pages 82–91, Philadelphia.

A.W. Moore and M.S. Lee. 1994. Efficient algorithms for minimizing cross validation error. In *International Conference on Machine Learning"*, pages 190–198, New Brunswick.

H.T. Ng and H.B. Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An examplar-based approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL-96)*, Santa Cruz.

T. Pedersen. 2001. A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings of the North American Chapter of the Association for Compuatational Linguistics, NAACL 2001*, pages 79–86, Pittsburg.

M. Stevenson and Y. Wilks. 2001. The interaction of knowledge sources in word sense disambiguation. *Computational Linguistics*, 27(3):321–351.

J. Veenstra, A. van den Bosch, S. Buchholz, W. Daelemans, and J. Zavrel. 2000. Memory-based word sense disambiguation. *Computers and the Humanities*, 34:171–177.

D. Yarowsky. 2000. Hierarchical decision lists for word sense disambiguation. *Computers and the Humanities*, 34:179–186.