Word Sense Disambiguation using Machine Learning Techniques

Gerard Escudero Bakx

Advisors:
Lluís Màrquez Villodre and German Rigau Claramunt

Universitat Politècnica de Catalunya

July 13th, 2006

Summary

- Introduction
- Comparison of ML algorithms
- Domain dependence of WSD systems
- Bootstrapping
- Senseval evaluations at Senseval 2 and 3
- Conclusions

Word Sense Disambiguation

sense	gloss from WordNet 1.5
age 1	the length of time something (or someone) has existed
age 2	a historic period

He was mad about stars at the **age** of nine .

WSD has been defined as AI-complete (Ide & Véronis, 1998);
such as the representation of world knowledge

Usefulness of WSD

- WSD is a potential *intermediate task* (Wilks & Stevenson, 1996) for many other NLP systems
- WSD capabilities appears in many applications:
  - ⋆ Machine Translation (Weaver, 1955; Yngve, 1955; Bar-Hillel, 1960)
  - ⋆ Information Retrieval (Salton, 1968; Salton & McGill, 1983; Krovetz & Croft, 1992; Voorhees, 1993; Schütze & Pedersen, 1995)
  - ⋆ Semantic Parsing (Alshawi & Carter, 1994)
  - ⋆ Speech Synthesis and Recognition (Sproat et al., 1992; Yarowsky, 1997; Connine, 1990; Seneff, 1992)
  - ⋆ Natural Language Understanding (Ide & Véronis, 1998)
  - ⋆ Acquisition of Lexical Knowledge (Ribas, 1995; Briscoe & Carroll, 1997; Atserias et al., 1997)
  - ⋆ Lexicography (Kilgarriff, 1997)
- Unfortunately, this usefulness has still not been demonstrated

WSD approaches

- all approaches build a model of the examples to be tagged
- according to the source of the information they use to build this model, systems can be classified as:
  - ⋆ knowledge-based: information from a external knowledge source, like a machine-readable dictionary or a lexico-semantic ontology
  - ⋆ corpus-based: information from examples
    - ∗ supervised learning: when these examples are labelled with its appropriate sense
    - ∗ unsupervised learning: when the examples have no sense information

Corpus-based and Machine Learning

- most of the algorithms and techniques to build models from examples (corpus-based) come from the Machine Learning area of AI

- WSD as a classification problem:

  ⋆ senses are the classes
  ⋆ examples should be represented as features (or attributes)
    ∗ local context: i.e. word at right position is a verb
    ∗ topic or broad-context: i.e. word "years" appears in the sentence
    ∗ syntactical information: i.e. word "ice" as noun modifier
    ∗ domain information: i.e. example is about "history"

- supervised methods suffer the "knowledge acquisition bottleneck" (Gale et al., 1993)

  ⋆ the lack of widely available semantically tagged corpora, from which to construct really broad coverage WSD systems, and the high cost in building one

"Bottleneck" research lines

- automatic acquisition of training examples
  - ⋆ an external lexical source (i.e. WordNet) or a seed sense-tagged corpus is used to obtain new examples from an untagged very large corpus or the web (Leacock et al., 1998; Mihalcea & Moldovan, 1999b; Mihalcea, 2002a; Agirre & Martínez, 2004c)
- active learning
  - ⋆ is used to choose informative examples for hand tagging, in order to reduce the acquisition cost (Argamon-Engelson & Dagan, 1999; Fujii et al., 1998; Chklovski & Mihalcea, 2002)
- bootstrapping
  - ⋆ methods for learning from labelled and unlabelled data (Yarowsky, 1995b; Blum & Mitchell, 1998; Collins & Singer, 1999; Joachims, 1999; Dasgupta et al., 2001; Abney, 2002; 2004; Escudero & Màrquez, 2003; Mihalcea, 2004; Suárez, 2004; Ando & Zhang, 2005; Ando, 2006)
- semantic classifiers vs word classifiers
  - ⋆ building of semantic classifiers by merging training examples from words in the same semantic class (Kohomban & Lee, 2004; Ciaramita & Altun, 2006)

Other active research lines

- automatic selection of features
  - ⋆ sensitiveness to non relevant and redundant features (Hoste et al., 2002b; Daelemans & Hoste, 2002; Decadt et al., 2004)
  - ⋆ selection of best feature set for each word (Mihalcea, 2002b; Escudero et al., 2004)
  - ⋆ to adjust the desired precision (at the cost of coverage) for high precision disambiguation (Martínez et al., 2002)
- parameter optimisation
  - ⋆ using Genetic Algorithms (Hoste et al., 2002b; Daelemans & Hoste, 2002; Decadt et al., 2004)
- knowledge sources
  - ⋆ combination of different sources (Stevenson & Wilks, 2001; Lee et al., 2004)
  - ⋆ different kernels for different features (Popescu, 2004; Strapparava et al., 2004)

Supervised WSD approaches by induction principle

- probabilistics models
  - ⋆ Naive Bayes (Duda & Hart, 1973): (Gale et al., 1992b; Leacock et al., 1993; Pedersen and Bruce, 1997; Escudero et al., 2000d; Yuret, 2004)
  - ⋆ Maximum Entropy (Berger et al., 1996): (Suárez and Palomar, 2002; Suárez, 2004)
- similarity measures
  - ⋆ VSM: (Schütze, 1992; Leacock et al., 1993; Yarowsky, 2001; Agirre et al., 2005)
  - ⋆ kNN: (Ng & Lee, 1996; Ng, 1997a; Daelemans et al., 1999; Hoste et al., 2001; 2002a; Decadt et al., 2004, Mihalcea & Faruque, 2004)
- discriminating rules
  - ⋆ Decision Lists: (Yarowsky, 1994; 1995b; Martínez et al., 2002; Agirre & Martínez, 2004b)
  - ⋆ Decision Trees: (Mooney, 1996)
  - ⋆ Rule combination, AdaBoost (Freund & Schapire, 1997): (Escudero et al., 2000c; 2000a; 2000b)
- linear classifiers and kernel-based methods
  - ⋆ SNoW: (Escudero et al., 2000a)
  - ⋆ SVM: (Cabezas et al., 2001; Murata et al., 2001; Lee & Ng, 2002; Agirre & Martínez, 2004b; Escudero et al., 2004; Lee et al., 2004; Strapparava et al., 2004)
  - ⋆ Kernel PCA: (Carpuat et al., 2004)
  - ⋆ RLSC: (Grozea, 2004; Popescu, 2004)

Senseval evaluation exercises

- Senseval
  - ⋆ it was designed to compare, within a controlled framework, the performance of different approaches and systems for WSD (Kilgarriff & Rosenzweig, 2000; Edmonds & Cotton, 2001; Mihalcea et al., 2004; Snyder & Palmer, 2004)
  - ⋆ Senseval 1 (1998), Senseval 2 (2001), Senseval 3 (2004), SemEval 1 / Senseval 4 (2007)
- the most important tasks are:
  - ⋆ all words task: assigning the correct sense to all content words a text
  - ⋆ lexical sample task: assigning the correct sense to different occurrences of the same word
- Senseval classifies systems into two types: supervised and unsupervised
  - ⋆ knowledge-based systems (mostly unsupervised) can be applied to both tasks
  - ⋆ exemplar-based systems (mostly supervised) can participate preferably in the lexical-sample task

Main Objectives

- understanding the word sense disambiguation problem from the machine learning point of view
- study the machine learning techniques to be applied to word sense disambiguation
- search the problems that should be solved in developing a broad-coverage and high accurate word sense tagger

Summary

- Introduction
- Comparison of ML algorithms
- Domain dependence of WSD systems
- Bootstrapping
- Senseval evaluations at Senseval 2 and 3
- Conclusions

Setting

- 10-fold cross-validation comparison
- paired Student's t-test (Dietterich, 1998) (with $t_{9,0.995} = 3.250$)
- data from DSO corpus (Ng and Lee, 1996)
- 13 nouns (*age, art, body, car, child, cost, head, interest, line, point, state, thing, work*) and 8 verbs (*become, fall, grow, lose, set, speak, strike, tell*)
- set of features:
  - ⋆ local context: w_{-1}, w_{+1}, (w_{-2}, w_{-1}), (w_{-1}, w_{+1}), (w_{+1}, w_{+2}), (w_{-3}, w_{-2}, w_{-1}), (w_{-2}, w_{-1}, w_{+1}), (w_{-1}, w_{+1}, w_{+2}), (w_{+1}, w_{+2}, w_{+3}), p_{-3}, p_{-2}, p_{-1}, p_{+1}, p_{+2}, and p_{+3}
  - ⋆ broad context information (bag of words): $c_1 \ldots c_m$

Algorithms Compared

- Naive Bayes (NB)
  - ⋆ positive information (Escudero et al., 2000d)
- Exemplar-based (kNN)
  - ⋆ positive information (Escudero et al., 2000d)
- Decision Lists (DL) (Yarowsky, 1995b)
- AdaBoost.MH (AB)
  - ⋆ LazyBoosting (Escudero et al., 2000c)
  - ⋆ local features binarised and topical as binary test (from 1,764 to 9,990 features)
- Support Vector Machines (SVM)
  - ⋆ linear kernel and binarised features

Adaptation Starting Point

- Mooney (1996) and Ng (1997a) were two of the most important comparisons in supervised WSD previous the edition of Senseval (1998)

- both works contain contradictory information

Mooney	Ng
$NB > EB$	$EB > NB$
more algorithms	more words
EB with Hamming metric richer feature set	EB with MDVM metric only 7 feature types

- another surprising result is that the accuracy of (Ng, 1997a) was 1-1.6% higher than (Ng & Lee, 1996) with a poorer set of attributes under the same conditions

Improving EB and NB (Escudero et al., 2000d)

- broad-context attributes with Exemplar Based (EB):
  - ⋆ sparse vector representation $\longrightarrow$ vector of about 5,000 0's and only a few 1's
  - ⋆ examples coincide in "all" zeros and biases the similarity measure to the shortest sentences
  - ⋆ it explains poor results of kNN in Mooney (1996), and better results of simpler feature sets in (Ng & Lee, 1996; Ng, 1997a)
  - ⋆ solution (PEB): the similarity function as the number of words shared $S(V_i, V_j) = \| V_i \cap V_j \|$ where $V_k = \{w_{k_1}, w_{k_2}, \ldots, w_{kn}\}$ is the set of content words in the sentence
- Naive Bayes (NB):
  - ⋆ PNB: combine only the conditional probabilities of the words appearing in test examples
- conclusions:
  - ⋆ PEB improves by 10 points accuracy of EB and is 15 times faster
  - ⋆ PNB is at least as accurate as NB and is 80 times faster
  - ⋆ exemplar–Based approach is better in accuracy than Naive Bayes

Accelerating AdaBoost.MH (Escudero et al., 2000c)

- AdaBoost.MH is an iterative process that builds a classifier by combining weak classifiers
- there is an initial distribution of weights: each example has the same value
- at each iteration:
  - ⋆ the best feature is selected to construct a rule (weak rule)
  - ⋆ weight of examples with good (or bad) predictions are decreased (or increased)
- LazyBoosting: at each iteration the best feature is selected among a small proportion p of randomly selected features
  - ⋆ this proportion p is recalculated at each iteration
  - ⋆ dropping the 90% of examples at each iteration, LazyBoosting achieves the same performance and runs about 7 times faster than AdaBoost.MH

Comparison Accuracy Results

accuracy and standard deviation of all learning methods:

	MFC	NB	kNN	DL	AB	SVM
nouns	46.59±1.08	62.29±1.25	63.17±0.84	61.79±0.95	66.00±1.47	66.80±1.18
verbs	46.49±1.37	60.18±1.64	64.37±1.63	60.52±1.96	66.91±2.25	67.54±1.75
all	46.55±0.71	61.55±1.04	63.59±0.80	61.34±0.93	66.32±1.34	67.06±0.65

- all methods outperform MFC (from 15 to 20.5 points)
- best systems are AB and SVM, and worst systems are NB and DLs
- regarding standard deviation, SVM has shown the most stable behaviour

$$MFC < DLs \approx NB < kNN < AB \approx SVM$$

SVM vs AB and number of training examples

- AdaBoost can perform badly using small training sets (Schapire, 2002)

- overall accuracy of AB and SVM classifiers by groups of words of increasing average number of examples per sense:

	<35	**35-60**	**60-120**	**120-200**	**>200**
AB	60.19%	57.40%	70.21%	65.23%	73.93%
SVM	63.59%	60.18%	70.15%	64.93%	72.90 %

  - ⋆ $SVM > AB$ when $\#examples < 60$
  - ⋆ $SVM < AB$ when $\#examples > 120$

Agreement and Kappa

kappa statistic (Cohen, 1960) (below diag.) and percentage of agreement (above diag.):

	DSO	MFC	NB	kNN	DL	AB	SVM
DSO	–	46.6	61.5	63.6	61.3	66.3	67.1
MFC	-0.19	–	73.9	58.9	64.9	54.9	57.3
NB	0.24	-0.09	–	75.2	76.7	71.4	76.7
kNN	0.39	-0.15	0.43	–	70.2	72.3	78.0
DL	0.31	-0.13	0.39	0.40	–	69.9	72.5
AB	0.44	-0.17	0.37	0.50	0.42	–	80.3
SVM	0.44	-0.16	0.49	0.61	0.45	0.65	–

- NB is the most similar to MFC
- SVM and AB have the best learning behaviour
  - ⋆ most similar to DSO, and less similar to MFC
- NB and DLs achieve similar accuracy results, but their predictions are quite different
- kappa values of 0.44 for AB and SVM could be good
  - ⋆ Ng et al., (1999) report 56.7% accuracies and kappa values of 0.317 by humans

Comparison Conclusions

- contributions in the adaptation of 3 ML algorithms
  - ⋆ accuracy improvement and acceleration of Exemplar Based by feature representation
  - ⋆ acceleration of Naive Bayes by feature representation
  - ⋆ clarification confusing information between (Mooney, 1996) and (Ng & Lee, 1996; Ng, 1997a)
  - ⋆ acceleration of AdaBoost.MH (called LazyBoosting)
- a comparison between five of the State-of-the-Art algorithms in WSD has been performed
  - ⋆ accuracy, agreement and kappa values and number of training examples (SVM and AB)
- SVM and AB are the best learning algorithms for the task
  - ⋆ SVM for small training sets (less than 60 examples per sense)
  - ⋆ AB for larger sets (more than 120 examples per sense)
  - ⋆ up to now, no larger sets are available; so, probably the best is SVM

Summary

- Introduction
- Comparison of ML algorithms
- Domain dependence of WSD systems
- Bootstrapping
- Senseval evaluations at Senseval 2 and 3
- Conclusions

Overview

- supervised learning assumes that training examples are somehow reflective of the task
  - ⋆ performance of systems is commonly estimated by testing on examples different from training, but they belong to the same corpus, and, therefore, they are expected to be quite similar
  - ⋆ Ng (1997a) estimates that the manual annotation effort necessary to build a broad coverage semantically annotated English corpus in about 16 person-years
  - ⋆ this methodology could be valid for some NLP problems, such as Part-of-Speech tagging

- we think that there exists no reasonable evidence to say that, in WSD, accuracy results cannot be simply extrapolated to other domains (Gale et al.,1992; Ng & Lee, 1996; Ng, 1997a; Martínez & Agirre, 2000)

- we have empirically studied the domain dependence dimension (Escudero et al., 2000a; 2000b), such as Sekine (1997) has done for parsing

Setting

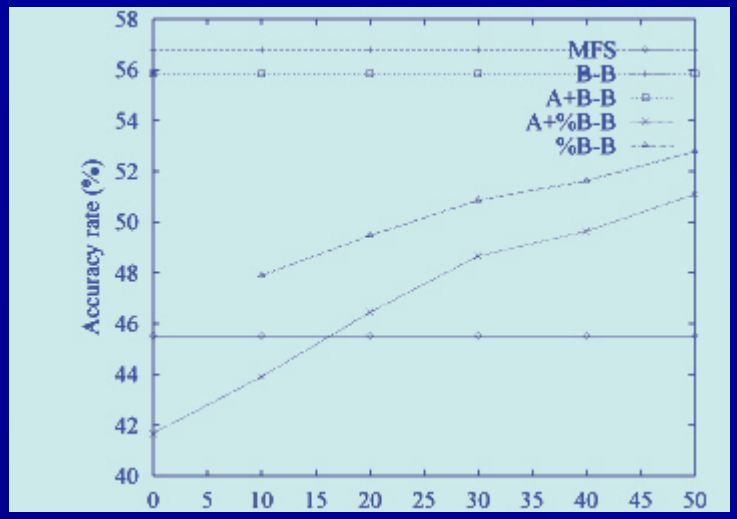
- DSO corpus (Ng and Lee, 1996)
  - ⋆ WSJ: from the financial domain (hereinafter A)
  - ⋆ BC: general corpus (hereinafter B)
- 13 nouns and 8 verbs (same words as in previous comparison)
- local (forms and part-of-speech) and topic information (bag of words)
- 4 ML methods: Naive Bayes, Exemplar Based, Decision Lists, and LazyBoosting
- McNemar's test ($\chi^2_{1,0.95} = 3.842$) and paired Student's t-test ($t_{9,0.975} = 2.262$) of significance

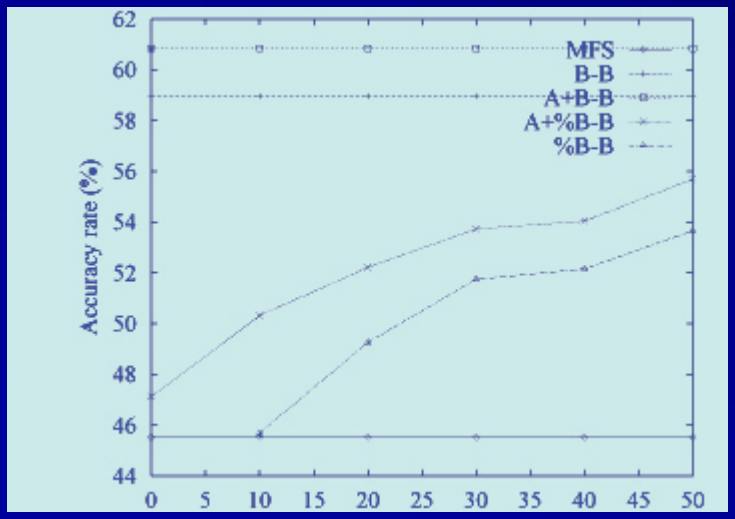
Across Corpora Evaluation

	MFC	NB	EB	DL	LB
$A + B \rightarrow A + B$	46.55	61.55	63.01	61.58	66.32
$A + B \rightarrow A$	53.90	67.25	69.08	67.64	71.79
$A + B \rightarrow B$	39.21	55.85	56.97	55.53	60.85
$A \rightarrow A$	55.94	65.86	68.98	67.57	71.26
$B \rightarrow B$	45.52	56.80	57.36	56.56	58.96
$A \rightarrow B$	36.40	41.38	45.32	43.01	47.10
$B \rightarrow A$	38.71	47.66	51.13	48.83	51.99

- **LB** outperforms all methods
- the addition of extra examples does not contribute to improve:
  $A + B \rightarrow A$ is similar to $A \rightarrow A$
  $A + B \rightarrow B$ is similar to $B \rightarrow B$
- very disappointing results are obtained regarding portability:
  $A \rightarrow B$ is 12 points under $B \rightarrow B$ for **LB**
  $B \rightarrow A$ is 19 points under $A \rightarrow A$ for **LB** and under the **MFC**

Tuning to the target corpus



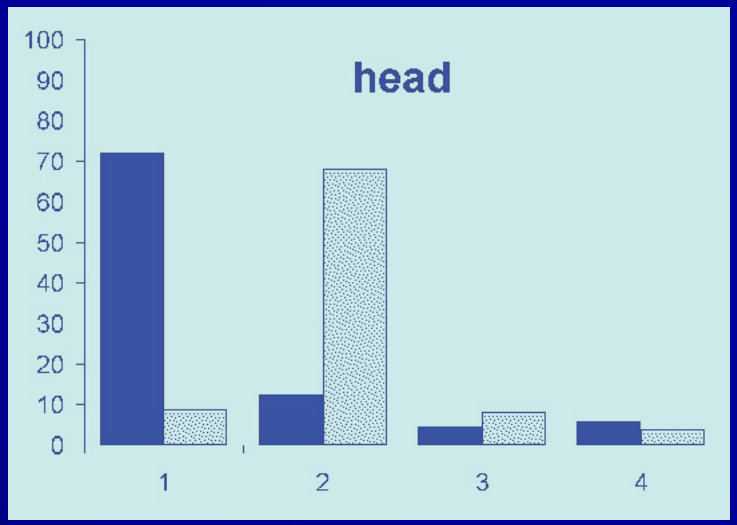


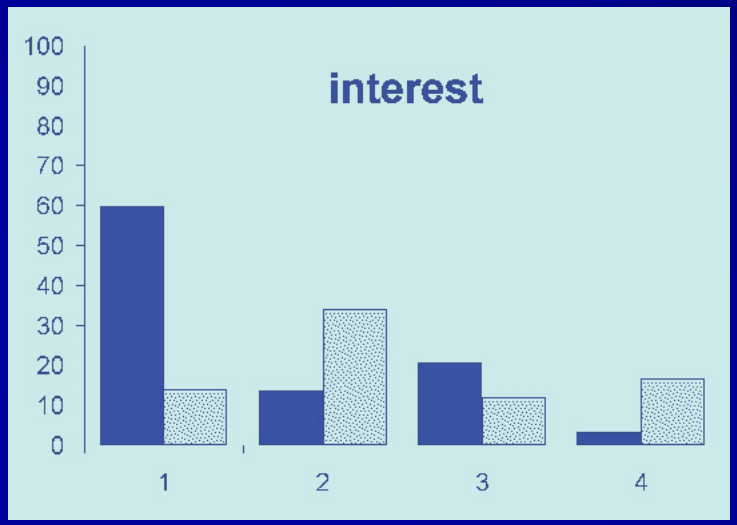
- in 3 cases the contribution is almost negligible: **NB** on B, **EB** on A, and **DL** on B
- in 3 cases a degradation on the accuracy is observed: **NB** on A (left picture), **EB** on B, and **DL** on A
- in both LazyBoosting cases a moderate improvement is observed (right picture)

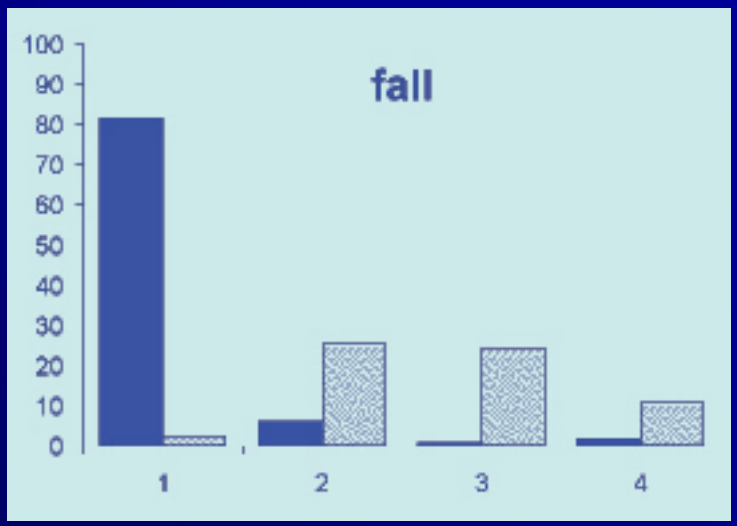
Training Data Quality

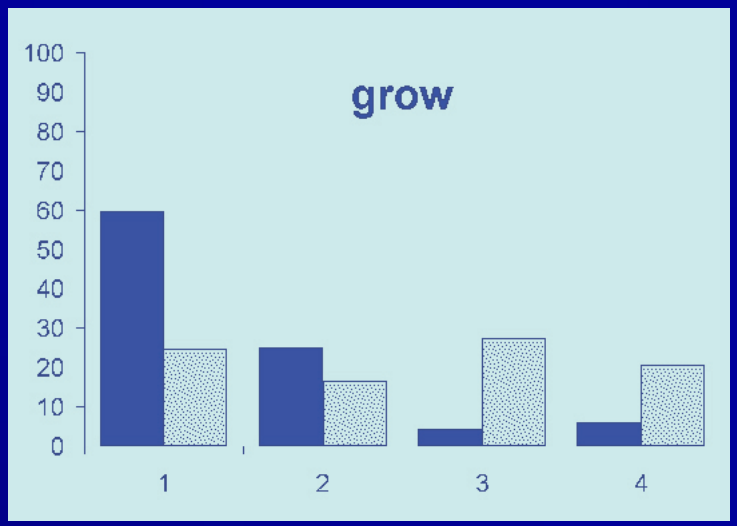
bad portability results may be, at least, explained by:

1. different sense distribution in both corpora









Training Data Quality (II)

2. examples contain different information

	MFC	LB
$A + B \rightarrow A + B$	48.55	64.35
$A + B \rightarrow A$	48.64	66.20
$A + B \rightarrow B$	48.46	62.50
$A \rightarrow A$	48.62	65.22
$B \rightarrow B$	48.46	61.74
$A \rightarrow B$	48.70	56.12
$B \rightarrow A$	48.70	58.05

sense balanced corpora

		senses					
	source	0	1	2	3	4	5
rule 1	A	-0.5064	-1.0970	-0.6992	1.2580	-1.0125	-0.4721
		0.0035	0.0118	0.0066	-0.0421	0.0129	0.0070
rule 2	B	-0.0696	-0.2988	-0.1476	-1.1580	-0.5095	1.2326
		-0.0213	0.0019	-0.0037	0.0102	0.0027	-0.0094

manual inspection of the rules learned by **LB** (collocation *state court*)

Domain Dependence Conclusions

- this work has pointed out major difficulties regarding the portability of supervised WSD systems
  - ⋆ the performance of supervised sense taggers is not guaranteed when moving from one domain to another
  - ⋆ some kind of adaptation is required
- these results are in conflict with the idea of "robust broad-coverage WSD" (Ng, 1997b)
- AdaBoost has better properties when is applied to new domains than the other three methods

Summary

- Introduction
- Comparison of ML algorithms
- Domain dependence of WSD systems
- Bootstrapping
- Senseval evaluations at Senseval 2 and 3
- Conclusions

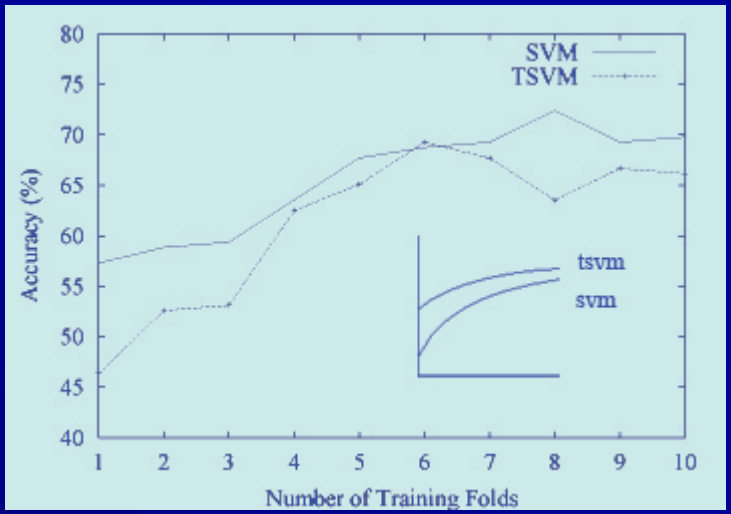
Overview

- bootstrapping is one of the most challenging open lines of research in the field (Agirre et al., 2005)
- this section explores the usefulness of unlabelled examples in Supervised WSD systems
- we explore two approaches:
  - ⋆ *Transductive* SVMs (Joachims, 1999)
  - ⋆ *Greedy Agreement* algorithm (Abney, 2002; 2004; Collins & Singer, 1999)
    - ∗ the application of this algorithm to WSD is at an initial stage

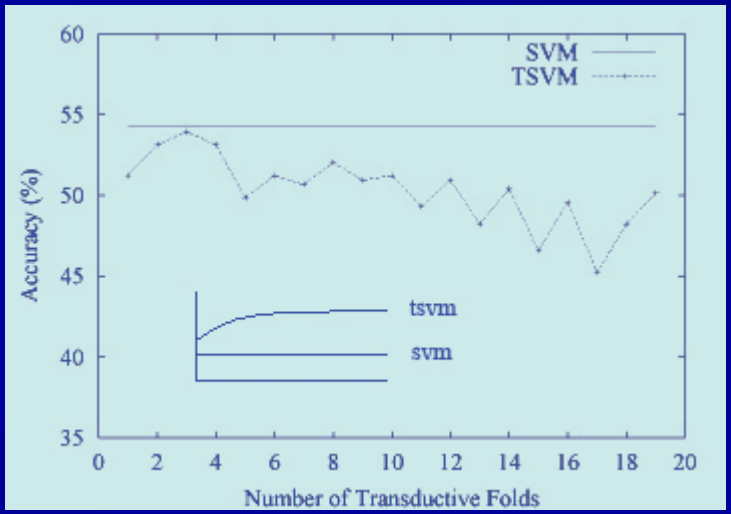
Transductive SVMs (Escudero & Màrquez, 2003)

- repeating the experiment of Joachims (1999) from Text Categorisation to Word Sense Disambiguation
- 4 nouns (*art*, *law*, *state*, and *system*) and 4 verbs (*become*, *leave*, *look*, and *think*) were selected from DSO

training variation



test variation



noun *art*

- the learning curves are very disappointing
- conclusion: with this setting, the use of unlabelled examples with the *transductive approach* is not useful for WSD
  - ⋆ it has to be studied the differences between Text Categorisation and WSD

Greedy Agreement Algorithm

- needs a seed as start point: rules with high precision but low coverage
- iterative process: at each iteration a rule (based on a feature) is obtained
  - ⋆ features are not discarded
  - ⋆ the number of times a feature is selected acts as a kind of weight
- two views (*conditionally independent* given the class value): the iterative process learns rules of both views alternatively
- the stopping criterion is defined by a cost function (function of both labelled and unlabelled examples)

Adapting Greedy Agreement to WSD

- seeds:
  - ⋆ training high accurate (but low coverage) Decision Lists classifiers
  - ⋆ selection of the rules with a weight over a fixed threshold

- views:

view	feature	description
F	bigsf	word form bigrams
F	fcomb	word forms pairs (consecutive or not) of the open-class-words
F	top_f	bag of word forms
G	f_loc	word forms and position in a ± 3-word-window
G	form	word form of the target word
G	p_loc	part-of-speech an position in a ± 3-word-window

Greedy Agreement Setting

- first experiment in porting the experiment of Abney (2000) from Name Entity Recognition to WSD
- 2 nouns (*law* and *system*) and 2 verbs (*become* and *think*) were selected from DSO

	senses		**examples**	
law.n	14:00	10:00	444	361
system.n	06:00	09:02	417	141
become.v	30:00	42:00	606	532
think.v	31:01	31:00	690	429

- 10 folds have been built:
  - ⋆ fold 0: seed
  - ⋆ folds 1–8: training
  - ⋆ fold 9: test

Greedy Agreement results

- results on the test set $(\mathbf{0} \longrightarrow (\mathbf{1} - \mathbf{8}) \longrightarrow \mathbf{9})$ (F1 cov.):

	law.n	system.n	become.v	think.v
Baselines				
MFC	55.2 100	74.7 100	53.3 100	61.7 100
DLs	44.3 72.4	59.7 71.8	81.1 94.5	85.2 97.4
LB	64.5 100	71.8 100	85.3 100	86.7 100
only seed rules				
F	53.4 92.1	59.3 73.1	54.2 42.2	70.1 86.7
G	57.8 68.4	33.0 32.1	72.5 77.1	85.3 86.7
priority_F	57.3 97.4	62.9 79.5	75.6 84.4	81.2 83.2
priority_G	61.3 97.4	57.1 79.5	77.6 84.4	77.3 99.1
Greedy Agreement				
Iterations	93	114	93	101
Threshold	2	2.5	3	2
F	61.5 88.2	67.1 87.2	73.1 70.6	79.1 94.7
G	64.7 75.0	42.8 68.0	78.0 83.5	83.6 88.5
priority_F	64.0 93.4	67.1 94.9	85.3 93.6	82.3 100
priority_G	72.1 93.4	59.2 94.9	83.4 93.6	88.5 100

- in 2 cases the results are better, in one case they are equivalent and in one case the best system is the MFC
- the coverage is over 93%
- the best threshold value is different for each word

Greedy Agreement conclusions

- the Greedy Agreement algorithm is providing interesting but not conclusive results
  - ⋆ 90,000 NER vs 500 WSD unlabelled examples
  - ⋆ almost 100% NER vs 60-90% seed accuracy

Bootstrapping current and further work

- testing with all senses (not only 2 of them)
- increasing the feature set
- optimising the seed threshold by a cross-validation framework
- testing the algorithm with large sets of unlabelled data from BNC

Summary

- Introduction
- Comparison of ML algorithms
- Domain dependence of WSD systems
- Bootstrapping
- Senseval evaluations at Senseval 2 and 3
- Conclusions

English Lexical Sample data in Senseval

- Senseval-2 English Lexical Sample
  - ⋆ examples: 8,611 for training and 4,328 for test of 73 words
  - ⋆ senses:
    - ∗ regular senses:

art regular senses			
sense	**word**	**pos**	**descriptor**
art%1:04:00::	art	noun	creation
art%1:06:00::	art	noun	artworks
art%1:09:00::	art	noun	skill
art%1:10:00::	art	noun	inabook

    - ∗ special cases: proper-nouns, multiwords, phrasal verbs, and unassignable
- Senseval-3 English Lexical Sample
  - ⋆ examples: 7,860 for training and 3,944 for test of 57 words
  - ⋆ senses: regular and unassignable

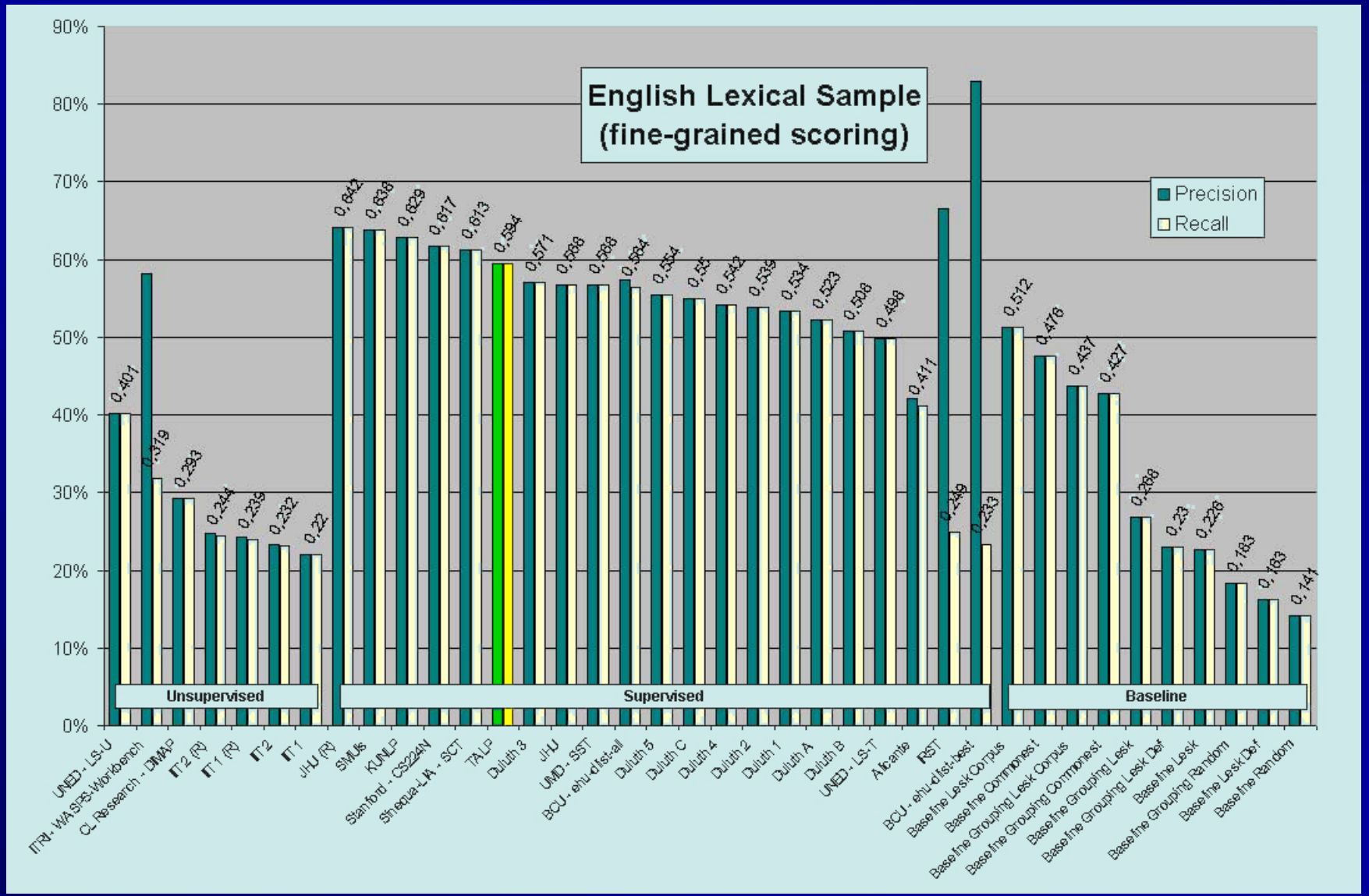
English Lexical Sample task in Senseval-2

system characteristics (Escudero et al., 2001):

- LazyBoosting (variation of AdaBoost.MH) as learning algorithm
- external knowledge-based features (WordNet Domains) (Magnini & Cavaglia, 2000)
- a specific treatment of multiwords (storing those found in the training set)
- no treatment of proper-names neither of unassignable examples
- a hierarchical decomposition by semantic fields (Yarowsky, 2000)

English Lexical Sample task in Senseval-2

official results (available at http://www.senseval.org):



From Senseval-2 to Senseval-3

study of the results of the best six systems of Senseval-2:

system	fine	ok	coarse	ok	
JHU(R)	64.2	2,780	71.3	3,086	(Yarowsky et al., 2001)
SMUIs	63.8	2,763	71.2	3,080	(Mihalcea and Moldovan, 2001)
KUNLP	62.9	2,724	69.7	3,016	(Seo et al., 2001)
CS224N	61.7	2,670	68.9	2,981	(Ilhan et al., 2001)
Sinequa	61.3	2,653	68.2	2,953	(Crestan et al., 2001)
TALP	59.4	2,571	67.1	2,903	(Escudero et al., 2001)
Examples: 4,328			Coverage: 100%		

- the predictions of all participant systems are available at Senseval web page (http://www.senseval.org)

Kappa and agreement of Senseval 2 systems

	jhu	smuis	kunlp	cs224n	sinequa	talp
jhu	-	0.62	0.67	0.65	0.64	0.61
smui	0.34	-	0.66	0.60	0.64	0.60
kunlp	0.35	0.34	-	0.68	0.67	0.64
cs224n	0.29	0.22	0.31	-	0.67	0.65
sinequa	0.28	0.31	0.31	0.25	-	0.64
talp	0.27	0.27	0.30	0.25	0.26	-

- the low agreements (from 0.6 to 0.68) and kappa values (from 0.22 to 0.35) show the significant differences in the output predictions

Proper Names:

system	pr	rc	F1	ok	att
JHU(R)	54.8	34.9	42.6	51	93
SMUIs	81.6	48.6	60.9	71	87
KUNLP	N/A	0	N/A	0	0
CS224N	66.7	1.4	2.7	2	3
Sinequa	88.9	5.5	10.4	8	9
TALP	N/A	0	N/A	0	0
Total: 146					

similar tables have been obtained for:

- multiwords
- Phrasal Verbs
- unassignable

Study conclusions

- the intersection set of all the examples in which every system have output a regular sense:

	accuracy	
Methods	**fine**	**coarse**
JHU(R)	65.5	72.9
SMUIS	65.2	72.9
KUNLP	66.0	73.4
CS224N	65.7	73.4
Sinequa	63.4	70.8
TALP	61.6	69.6
Total: 3,520		

- conclusions:
  - ⋆ all systems have achieved quite similar accuracies
  - ⋆ preprocessors were very important to construct a competitive WSD system at Senseval-2

The TalpSVM System

- following this study we implemented a new system:
  - ⋆ SVM due to the small size of training sets: SVM^{light} (Joachims, 2002)
  - ⋆ increase of the number of features
  - ⋆ improvement of the preprocessing components

system	fine	ok	coarse	ok
TalpSVM	64.8	2,806	71.7	3,105
JHU(R)	64.2	2,780	71.3	3,086
SMUIs	63.8	2,763	71.2	3,080
KUNLP	62.9	2,724	69.7	3,016
CS224N	61.7	2,670	68.9	2,981
Sinequa	61.3	2,653	68.2	2,953
TALP	59.4	2,571	67.1	2,903
Examples: 4,328				

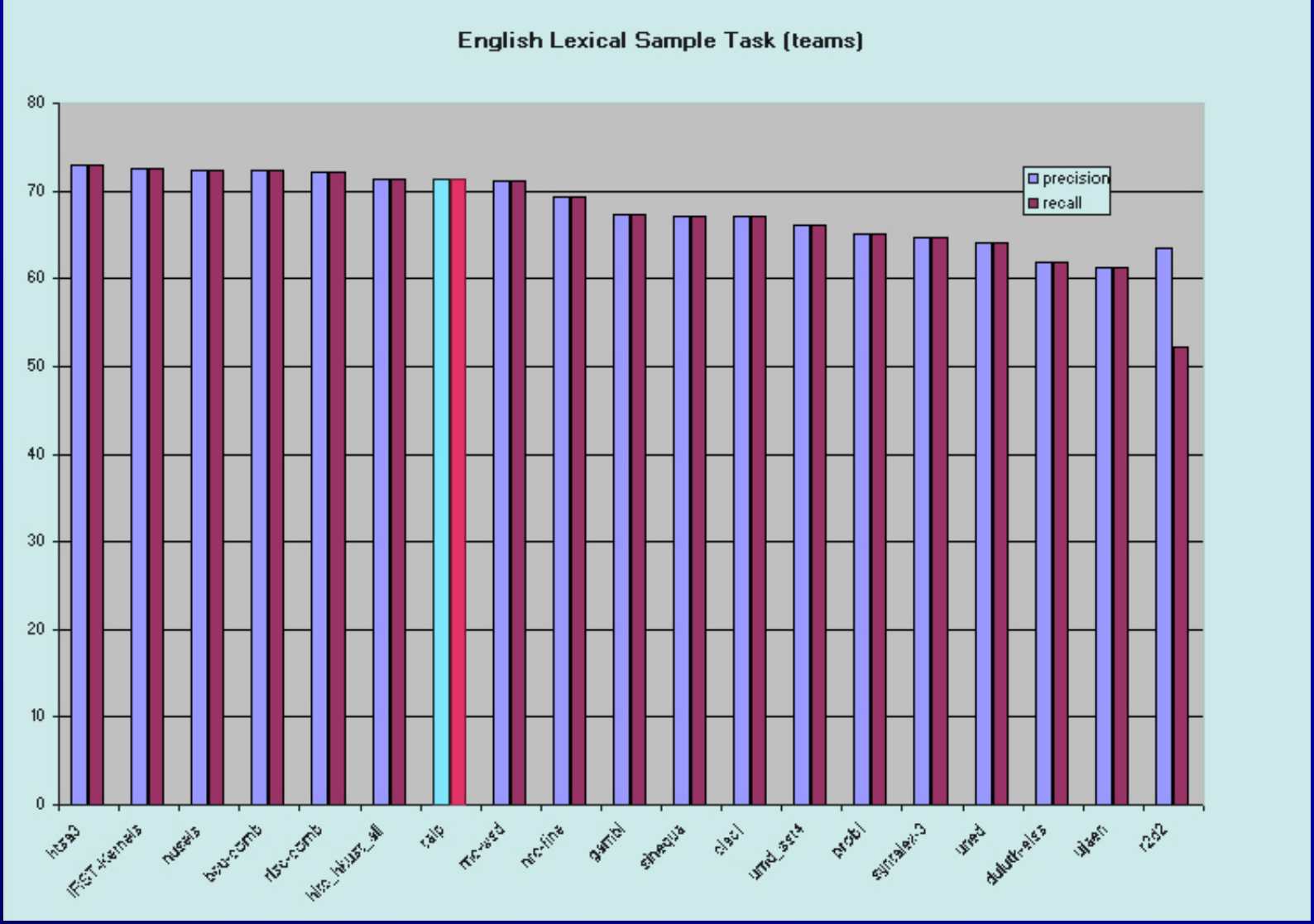
English Lexical Sample task in Senseval-3

system characteristics (Escudero et al., 2004):

- SVM as learning algorithm
- two binarisation procedures:
  - ⋆ one vs all
  - ⋆ constraint classification (Har-Peled et al., 2002)
- extended feature set:
  - ⋆ topical context
  - ⋆ local context
  - ⋆ syntactical relations: from Minipar (Lin, 1998) and patterns (software of Yarowsky's group)
  - ⋆ several domain information: semantic files, sumo ontology...
- a per-word feature and binarisation procedure selection

English Lexical Sample task in Senseval-3

official results:



Senseval conclusions

- we have participated in the last two editions of Senseval
  - ⋆ Senseval 2:
    - ∗ AdaBoost.MH
    - ∗ features increased with domain information
  - ⋆ Senseval 3:
    - ∗ Support Vector Machines
    - ∗ very rich and redundant features
    - ∗ two binarisation approaches: one vs all and constraint classification
    - ∗ per-word feature and binarisation selection
- our systems have achieved a very high position in both competitions
- we have performed a study of the best six systems of second edition
  - ⋆ we have shown the importance of preprocessors in Senseval 2

Summary

- Introduction
- Comparison of ML algorithms
- Domain dependence of WSD systems
- Bootstrapping
- Senseval evaluations at Senseval 2 and 3
- Conclusions

Contributions

- comparison:
  - ⋆ improvement in accuracy and/or efficiency of 3 ML algorithms: PEB, PNB and LB
    - ∗ clarification of some confusing information on the literature
  - ⋆ many dimensions comparison of 5 ML algorithms: NB, EB, DL, AB, and SVM
    - ∗ best systems are margin maximisation classifiers
- domains:
  - ⋆ empirically showing the dependence of the ML classifiers to the training data
- bootstrapping:
  - ⋆ application of *Transductive SVM*
  - ⋆ first attempts of *Greedy Agreement* algorithms to WSD
- Senseval participation:
  - ⋆ participation achieving good results in two editions of Senseval
  - ⋆ comparative study of six best systems of Senseval 2

Conclusions

- corpus-based approaches have obtained the best absolute results on WSD
- systems are still not useful for practical purposes
  - ⋆ best systems obtained accuracy lower than 75%
- Senseval results show all systems are quite similar
  - ⋆ we need somehow a *qualitative* precision increase
- the only taggers we can build are those based on particular domains
  - ⋆ until knowing the way of building general corpora, training less dependant classifiers, or adapting classifiers to new corpora
- before stating that the supervised ML paradigm is able to resolve a realistic WSD problem
  - ⋆ there is a list of open problems regarding: portability, tuning, knowledge acquisition, accuracy, etc
- the most important outstanding problem is the demonstration, on a real application, of the usefulness of WSD

Further Work

- the application of bootstrapping techniques to Word Sense Disambiguation
- the application of *on-line* learning classifiers able to adapt to new domains
- the representation of examples as features
- the integration of WSD classifiers within other NLP tasks (such as semantic parsing), or opportunity multi-task semantics (Ando, 2006)
- the participation in next edition of Senseval (2007)

Selected Publications

comparison of supervised ML algorithms for WSD:

G. Escudero, L. Màrquez, and G. Rigau. Boosting Applied to Word Sense Disambiguation. In *Proceedings of the 12th European Conference on Machine Learning, ECML*, 2000.

G. Escudero, L. Màrquez, and G. Rigau. Naive Bayes and Exemplar-Based Approaches to Word Sense Disambiguation Revisited. In *Proceedings of the 14th European Conference on Artificial Intelligence, ECAI*, 2000.

L. Màrquez, G. Escudero, D. Martínez, and G. Rigau. *Machine Learning Methods for WSD.* Chapter of the book *Word Sense Disambiguation: Algorithms, Applications, and Trends*. E. Agirre and P. Edmonds, editors. Kluwer, 2006.

cross-corpora evaluation and adaptation to new domains:

G. Escudero, L. Màrquez, and G. Rigau. A Comparison between Supervised Learning Algorithms for Word Sense Disambiguation. In *Proceedings of the 4th Computational Natural Language Learning Workshop, CoNLL*, 2000.

G. Escudero, L. Màrquez, and G. Rigau. An Empirical Study of the Domain Dependence of Supervised Word Sense Disambiguation Systems. In *Proceedings of the joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP/VLC*, 2000.

Selected Publications (II)

data scarceness: bootstrapping:

G. Escudero and L. Màrquez. *Transductive Approach using Support Vector Machines on Labelled and Unlabelled Data*. Deliverable WP6.5 of the MEANING Project (IST-2001-34460), 2003.

international evaluation exercises:

G. Escudero, L. Màrquez, and G. Rigau. Using Lazy Boosting for Word Sense Disambiguation. In *Proceedings of the 2nd International Workshop an Evaluating Word Sense Disambiguation Systems, Senseval-2*, 2001.

G. Escudero, L. Màrquez, and G. Rigau. TALP System for the English Lexical Sample Task. In *Proceedings of the 3rd International Workshop an Evaluating Word Sense Disambiguation Systems, Senseval-3*, 2004.