

ABSTRACT

My research at the University of Maryland involved three distinct but related areas within the field of computational linguistics:

- I. *Unsupervised morphological analysis (UMA)* – Breaking words into component morphemes for nearly any written language, using only raw text for analysis.
- II. *Word sense disambiguation (WSD)* – Distinguishing between a word's possible meanings.
 - A. Crosslingual word sense disambiguation (CWSD) – WSD where a word's meanings are identified by its translations into another language.
- III. *Machine translation (MT)* – Translating text automatically from one language to another, particularly making use of UMA and CWSD.

We evaluated the following experimental hypotheses:

- I. *Unsupervised morphological analysis (UMA)*
 - A. We can design a UMA system that applies to any morphologically complex language, and has accuracy comparable (not necessarily superior) to language-specific systems. (Task I)
- II. *Word sense disambiguation (WSD)*
 - B. WSD accuracy can be improved by providing unannotated data in addition to annotated training data. (Task II.A)

C. WSD accuracy can be improved by using document-level features in addition to sentence-level features. (Task II.B)

D. WSD accuracy can be improved by performing UMA on the text. (Task II.C)

III. *Machine translation (MT)*

E. MT accuracy can be improved by performing UMA on the source text.
(Task III.A)

F. MT accuracy can be improved by performing CWSD on the source text.
(Task III.B)

This paper summarizes all experimental results, with conclusions supporting hypotheses A, B, C and D.

Please note the partial glossary at the end of the paper.

Disclaimers: This paper is not intended for publication, but is rather a summary of my research efforts at UMD, from fall 2005 through 2006. It incorporates portions of several project reports, minimally edited, as well as descriptions of incomplete experiments. Summaries of related research efforts have generally not been updated since 2006.

**UNSUPERVISED MORPHOLOGICAL ANALYSIS AND
CROSSLINGUAL WORD SENSE DISAMBIGUATION
USING LARGE QUANTITIES OF UNANNOTATED DATA**

Edward Kenschaft

Master's thesis based on pre-candidacy doctoral research

University of Maryland

Copyright © 2008 Edward Kenschaft.

Table of Contents

1. Introduction.....	1
1.1. Goals.....	1
1.1.1. Task I: Unsupervised morphological analysis.....	1
1.1.2. Task II: Crosslingual word sense disambiguation.....	2
1.1.3. Task III: Machine translation.....	3
1.2. Related work.....	4
1.2.1. Related work: Unsupervised morphological analysis.....	4
1.2.1.1. Freitag (2005): Morphosyntactic transformation rules.....	4
1.2.1.2. Baldwin (2005): Deep lexical acquisition.....	5
1.2.2. Related work: Morphology & word sense disambiguation.....	6
1.2.3. Related work: Morphology & alignment / machine translation.....	6
1.2.3.1. Koehn & Knight (2003): German noun compounds.....	6
1.2.3.2. Nießen & Ney (2001a, 2004): Hierarchical lexicon.....	9
1.2.3.3. Popović & Ney (2004a, 2004b, 2005): Hierarchical lexicon.....	10
1.2.3.4. Schrader (2004): Stemming.....	11
1.2.3.5. Schrader (2006): Alignment of compound nouns with word sequences.....	12
1.2.3.6. Goldwater & McClosky (2005): Aligning morphologically rich languages.....	15
1.2.3.7. Lee (2004): Aligning templatic languages.....	21
1.2.3.8. Habash & Rambow (2005): Arabic morphological complexity.....	22
1.2.3.9. Isbihani et al. (2006): Segmenting Arabic.....	22
1.2.4. Related work: Crosslingual word sense disambiguation.....	23
1.2.4.1. Vickrey et al. (2005).....	23
1.2.5. Related work: WSD & machine translation.....	23
1.2.5.1. Carpuat & Wu (2005): Constrain MT using WSD.....	24
1.2.5.2. Cabezas & Resnik (2005): Improve MT using WSD.....	24
1.2.5.3. Pham et al. (2005).....	25
1.2.6. Related work: Phrase-based statistical machine translation (PBSMT).....	25
1.2.6.1. Chiang (2005): Hiero.....	25
1.2.7. Related work: Machine learning.....	26
1.2.7.1. Joachims (2003): Spectral Graph Transducer (SGT).....	26
1.2.8. Related work: Corpus linguistics.....	27
1.2.8.1. Babych & Hartley (2003): S-score.....	27
2. Task I: Unsupervised morphological analysis.....	28
2.1.1. Task formulation.....	28
2.1.1.1. Hypothesis.....	28
2.1.1.2. Objective.....	28
2.1.1.3. Languages of interest.....	29
2.1.1.4. Preprocessing methods.....	30
2.1.2. Evaluation.....	33

2.1.2.1.Complexity.....	33
2.1.2.2.Comparison against a standard.....	36
2.1.2.3.Downstream applications.....	37
3.Task II: Crosslingual word sense disambiguation.....	38
3.1.Task II.A: WSD using unlabeled data.....	38
3.1.1.Task formulation.....	39
3.1.1.1.Hypothesis.....	39
3.1.1.2.Software.....	39
3.1.1.3.Data.....	40
3.1.1.4.Feature Analysis.....	42
3.1.1.5.Evaluation.....	43
3.1.2.Results: WSD using unlabeled data.....	45
3.2.Task II.B: WSD using document-level features.....	50
3.2.1.Task formulation.....	50
3.2.1.1.Hypothesis.....	50
3.2.2.Results.....	50
3.3.Task II.C: WSD using morphological analysis.....	51
3.3.1.Task formulation.....	51
3.3.1.1.Hypothesis.....	51
3.3.1.2.Motivation.....	51
3.3.1.3.Data.....	51
3.3.2.Results.....	51
3.3.2.1.Results: Naive morphology.....	51
3.3.2.2.Results: Morphological analysis on Europarl data.....	54
3.3.2.3.Results: Classifier combination.....	55
4.Task III: Machine translation using morphological analysis and/or word sense disambiguation.....	58
4.1.Task III.A: Machine translation using morphological analysis.....	58
4.1.1.Task formulation.....	58
4.1.1.1.Hypothesis.....	58
4.1.1.2.Motivation.....	58
4.1.2.Results.....	59
4.2.Task III.B: Machine translation using word sense disambiguation.....	60
4.2.1.Task formulation.....	60
4.2.1.1.Hypothesis.....	60
4.2.1.2.Motivation.....	60
4.2.1.3.Platform.....	60
4.2.2.Results.....	61
4.2.2.1.WMT06.....	61
5.Summary.....	64
5.1.Conclusions.....	64
5.2.Future work.....	64

5.2.1.Future work related to morphological analysis.....	64
5.2.1.1.Future work related to morphological analysis heuristics.....	64
5.2.1.2.Future work related to morphological analysis evaluation.....	65
5.2.2.Future work related to word sense disambiguation.....	66
5.2.2.1.Future work related to WSD with unlabeled data.....	66
5.2.2.2.Future work related to WSD with document-level features.....	66
5.2.2.3.Future work related to WSD and morphological analysis.....	67
5.2.2.4.Other future work related to WSD.....	67
5.2.3.Future work related to machine translation.....	68
5.2.3.1.Future work related to MT with morphological analysis.....	68
5.2.3.2.Future work related to MT with WSD.....	68
5.2.3.3.Future work related to MT evaluation.....	69
6.Appendix: Additional resources.....	70
6.1.1.Rule-based morphological analysis.....	70
7.Glossary.....	72
7.1.1.1.Language types.....	81
8.References.....	82

Index of Tables

Table 1: Evaluation of compound-splitting methods (Koehn & Knight 2003).....	8
Table 2: Evaluation of syntactic restructuring methods (Nießen & Ney 2004)	10
Table 3: Corpus characteristics (Schrader 2006).....	13
Table 4: German compound length vs. English phrase length (Schrader 2006).....	13
Table 5: Morphological mappings: Czech-English (Goldwater & McClosky 2005).....	16
Table 6: Transformations (Goldwater & McClosky 2005).....	17
Table 7: BLEU scores for baseline & lemmatization.....	19
Table 8: BLEU scores (dev set) for method & class of tag.....	19
Table 9: BLEU scores for combined model.....	19
Table 10: BLEU-score evaluation of delete/merge (Lee 2004).....	22
Table 11: Rare events, by language, with and without morphological analysis.....	35
Table 12: F-score WSD results.....	45
Table 13: Naive morphology WSD results.....	53
Table 14: WSD results on Vickrey English data.....	56
Table 15: WMT06 results.....	63

Index of Figures

Figure 1: Rare word forms (Goldwater & McClosky 2005).....	16
------------------------------------------------------------	----

1. INTRODUCTION

1.1. Goals

1.1.1. Task I: Unsupervised morphological analysis

Morphological analysis (or *induction*) is the process of breaking a word, or all the words of a given language, into component morphemes. The field has a substantial history, but efforts have typically been constrained in major ways. Most efforts have focused on only a single language, rather than taking crosslingual patterns into account. Because of this, most of the developed techniques have been heavily language-dependent. Many efforts (i.e. rule-based or supervised statistical) have relied on extensive, time-consuming expert design or annotation. Evaluation standards are generally not obvious. Finally, until recent years, the task was rarely placed in the context of any larger goal, such as word sense disambiguation or translation.

Building upon this tradition, we sought to examine morphological patterns across a sampling of languages, and develop techniques that can be applied to (the regular morphology of) any language; to use exclusively unsupervised methods, requiring no expert rules or annotated data; and to evaluate results both independently and in terms of downstream applications (i.e. crosslingual word sense disambiguation and machine translation).

The goal was an unsupervised morphological analysis (UMA) system that can be used to induce morphological structure in any language where written text is available, regardless of what else is known about the language.

1.1.2. Task II: Crosslingual word sense disambiguation

Word sense disambiguation (WSD) is the process of determining which possible sense of a word is used in a given context. For instance, the English word *fence* has at least three senses:

1. He climbed over the *fence*.
2. He learned to *fence* with a sabre.
3. He went into the city to *fence* the stolen TV's.

Monolingual WSD is often a poorly defined problem. It is rarely obvious which sense inventory to use (Resnik & Yarowsky 2000). The required granularity of senses is subject to dispute, and may vary depending on the end purpose (Chugur et al. 2000). (For example, is "electric fence" the same sense as #1 above, or a distinct sense?) Even the definition of a "word" is inconsistent, sometimes taken to be an inflected word form, other times a root, an uninflected stem, or something else entirely.

Various researchers have begun to explore crosslingual word sense disambiguation (CWSD), based on a corpus of sentences that have been translated between two languages. The possible translations of a word in the target language are taken to be its

sense inventory. That is, if a word can be translated in three different ways, then these are its three "senses" – regardless of whether some other theoretical formalism would agree.

So far, research into CWSD has been constrained in several ways. Analysis has almost always been performed using sentence-level information only. Typically only manually aligned sentences are used, which means very little data is available. A "word" is generally taken to be an inflected word form, leading to even sparser data. The task has rarely been placed in the context of any larger goal, such as translation.

Building upon this tradition, we sought to introduce document-level features; to introduce large quantities of unannotated source-language data to reduce sparseness; use unsupervised morphological analysis to posit roots and affixes, again reducing sparseness; to train on automatically aligned (and hence noisy) data, again reducing sparseness; and to evaluate results both independently and in terms of a downstream application (i.e. machine translation).

Several of these approaches reduced data sparseness by introducing noisy data. Naturally, this involves trade-offs. However, results show that these approaches nonetheless helped in some cases. More research presumably could further improve noise filtering.

1.1.3. Task III: Machine translation

We attempted to evaluate UMA and CWSD in terms of their impact on machine translation (MT). Results are incomplete.

1.2. Related work

Disclaimer: These highlights do not scratch the surface of a full literature survey on these topics.

1.2.1. Related work: Unsupervised morphological analysis

Morphological analysis is the process of parsing a word into its component morphemes. This can be done manually or automatically, using either a rule-based or statistical system. Many of the most commonly studied languages have commercially available rule-based systems which achieve greater than 95% accuracy. However, developing such a system for a new language is time-intensive, as is annotating data for use by a supervised statistical system. We therefore focused our attention on unsupervised induction of morphological structure. Such a system could be created for a new language much faster than a rule-based or supervised system, as long as unannotated text is available.

This field has a long and rich history (e.g. Koskenniemi 1983; Kay 1987). We mention just a couple of efforts that most closely influenced our work.

1.2.1.1. Freitag (2005): Morphosyntactic transformation rules

Dayne Freitag (2005) offers a methodology for inferring morphosyntactic transformation rules, as opposed to morphemes per se. In a nutshell, his method involves the following steps:

1. Group all tokens into co-occurrence classes (*i.e. tokens that appear in similar contexts*).
2. Identify transformations between members of one co-occurrence class to members of another.
3. Cull transforms to a minimal set.

Some advantages of Freitag's system are that it provides a motivated distinction between roots and affixes; it rates affixes according to frequency, and allows a configurable admissibility threshold (*i.e. there must be at least k occurrences of an affix in order to be recognized*); and it can readily be adapted for more complicated morphosyntactic transformations, such as reduplication and deletion.

1.2.1.2. Baldwin (2005): Deep lexical acquisition

Morphological analysis is related to a task known in the literature as (deep) lexical acquisition (DLA). Timothy Baldwin (2005) assumes a seed database, and describes a variety of methods for augmenting that database. His method for identifying morphemes is summarized as follows:

1. Generate all 1- to 6-grams occurring in the corpus.
2. Filter out n -grams with fewer than ($m=3$) occurrences.
3. Filter out any n -gram with the same frequency as any supersequence (*i.e. a larger string in which it appears*).

4. Select the ($k=3900$) n -grams with the highest saturation (*i.e. most complete coverage of the data?*).

This method, I suspect, is theoretically inferior to Freitag's, since it does not share any of the advantages listed above, and arbitrarily limits the number of recognized morphemes. However, it should be significantly simpler and faster to implement.

Baldwin's non-morphological features depend on deeper semantic knowledge, which I do not presume to have.

1.2.2. Related work: Morphology & word sense disambiguation

Florian and Wicentowski (2002) used morphological analysis to improve WSD, referenced in (Wicentowski 2002).

I only discovered this paper recently (2008), and have not examined their work.

1.2.3. Related work: Morphology & alignment / machine translation

1.2.3.1. Koehn & Knight (2003): German noun compounds

Philipp Koehn & Kevin Knight (2003) experimented with heuristics for dividing German compounds into component morphemes, by looking for known roots that occur as substrings of larger words. They evaluated the effects of their approach on word alignment (German↔English), word-based statistical machine translation (WBSMT), and phrase-based statistical machine translation (PBSMT).

Because German is agglutinative, each German word (particularly noun compound) may align with an arbitrarily long sequence of English words. This provides a significant challenge for word alignment, and for other applications that depend upon alignment, such as machine translation.

Koehn & Knight attempted to address this problem by learning rules for splitting German noun compounds into component parts. A part is either a subword or filler between subwords. They used only unannotated text for training.

Their algorithm involved two steps: generation and selection.

For generation, they looked at a new word, and listed every possible covering of that word with known words and fillers. For example, if the new word is *aktionsplan*, the four possible coverings are:

1. *aktionsplan* (no covering)
2. *aktions-plan*
3. *aktion-s-plan*
4. *akt-ion-s-plan*

For selection, they employed various heuristics to determine which of these coverings is in some sense optimal. They experimented with a variety of different heuristics, including:

1. *eager* – split each word into the most (hence, smallest) possible chunks
2. *frequency-based* – employ a metric which takes into account frequency counts of various chunks in the training text
3. *parallel text* – use German↔English parallel text to test each proposed split against aligned text, keeping it if maps to an actual English phrase, discarding it if it does not
4. *part-of-speech* (POS) – restrict covering words to content words (e.g. not prepositions or determiners)

Their results are shown in Table 1.

Splitting method	Alignment			BLEU	
	precision	recall	accuracy	WBSMT	PBSMT
none	–	–	94.2%	29.1	30.5
eager	24.8%	73.3%	87.1%	22.2	34.4
frequency-based	57.4%	86.6%	95.7%	31.7	34.2
parallel	83.3%	89.1%	98.6%	29.4	33.0
parallel & POS	93.8%	90.1%	99.1%	30.6	32.6

Table 1: Evaluation of compound-splitting methods (Koehn & Knight 2003)

Not surprisingly, the best word alignment, 99.1% accuracy, occurred when using all possible information, parallel text plus POS, up from 94.2% with unmodified text. The frequency-based metric produced only produced moderate improvements, to 95.7%. Eager splitting hurt alignment.

It is perhaps surprising that the best downstream results with MT, as measured by BLEU score, did not involve optimization with parallel text. For word-based MT, the frequency-based metric was by far the best performer, with a 2.6 and 1.6 improvement in BLEU score, respectively, over no splitting and splitting based on all information.

Even more interesting are the results with phrase-based MT. Here the best approach was dumb, eager splitting, with a whopping 3.9 BLEU improvement over no splitting (34.4 vs. 30.5). The frequency-based metric was a close second (34.2).

The success of eager splitting is attributed to the peculiar strength of PBSMT, which has no problem taking an arbitrary string of "words" and regrouping them into "phrases" as needed.

1.2.3.2. Nießen & Ney (2001a, 2004): Hierarchical lexicon

Nießen & Ney (2001a, 2004) introduced the idea of a hierarchical lexicon, where a word is represented at various levels of inflectional specificity, starting with the bare root. German↔English alignment is performed at each of these levels. The technique was adopted by various later researchers.

Nießen & Ney also employed various heuristics for syntactic restructuring, such as reordering questions and detaching verb prefixes, to make the German more like the English. Finally, they added other heuristics, such as detecting multiword phrases based on POS sequence.

Used all together, the techniques led to improvements comparable to an order of magnitude of data (Table 2).

This claim would be stronger if an additional order of magnitude were represented, e.g. from 500 to 5K.

Each column represents a different accuracy measure; lower is better.

#sent	Method	1-BLEU	mWER	SSER	ISER
0	baseline	76.7	53.6	60.4	60.4
	restructure	70.9	50.2	57.8	30.0
	+ all	67.4	48.0	52.8	24.1
5K	baseline	52.6	38.0	37.3	17.4
	restructure	47.9	34.7	33.6	15.2
	+ all	47.1	33.9	31.8	13.7
	<i>baseline</i>	<i>46.3</i>	<i>34.1</i>	<i>30.2</i>	<i>14.1</i>
58K	restructure	43.7	32.5	26.6	12.8
	+ all	42.9	31.8	26.3	11.8

Table 2: Evaluation of syntactic restructuring methods (Nießen & Ney 2004)

1.2.3.3. Popović & Ney (2004a, 2004b, 2005): Hierarchical lexicon

Popović & Ney (2004a) built on the concept of the hierarchical lexicon (Nießen & Ney 2001a). They first parsed German words into roots and artificial inflectional morphemes, e.g. PRES for present tense. They then used a modified EM alignment algorithm to treat

each of these complex "words" as a hierarchy, with alignment possible at any level. They evaluated effects on a variety of baseline systems, producing at least modest improvements in every case.

Popović et al. (2004b, 2005) used knowledge of the specific language pair to remove inflectional morphemes or function words that are not translatable, leading to a significantly reduced lexicon, and reduced alignment error rate.

I presume the alignment of function words to NULL has been handled in the mainstream alignment literature, although I haven't read up on it. Removing inflectional morphemes by hand seems (a) ad hoc, and (b) potentially detrimental to translation due to loss of information. Splitting apart but retaining all morphemes should provide the same benefits, without these objections.

1.2.3.4. Schrader (2004): Stemming

Bettina Schrader (2004) also addressed German↔English alignment & translation (*among other things*), first looking at inflectional affixes, then noun compounding.

In experiments on three different data sets, Schrader removed inflectional affixes from the German, and aligned/translated only the lemmas. She evaluated on translation candidates generated using a bag-of-words dictionary generator (Hiemstra 1996). With all three data sets, recall went up with lemmatization. However, with the exception of one data set (*Patente*), precision went down. Schrader attributes this to a naive precision metric, which unduly penalizes the system for generating multiple translation candidates for the same term.

Using only the Patente data set, Schrader tried another experiment where she broke long compounds into component morphemes before alignment,

e.g. *Dämpfungsscheibenanordnung* ('dampening disk assembly') → *Dämpfung scheibe anordnung*. Recall improved somewhat, while precision did not change significantly.

Schrader noted that the system was severely hindered by its inability to handle alignments involving multi-word units. This suggests that results should be significantly better when evaluated according to a downstream PBSMT system.

1.2.3.5. Schrader (2006): Alignment of compound nouns with word sequences

Schrader (2006) addressed the problem of aligning German noun compounds with English word sequences. For data, she used the Europarl corpus (Koehn 2005) with both sides automatically POS tagged, and manually corrected alignments of ~100,000 German tokens to English counterparts.

Motivation

Schrader was particularly interested in the problems posed by an inordinately large proportion of hapax legomena (i.e. word forms occurring only once) in the German data, as hapax legomena cannot be handled neatly by a statistical system. As shown in Table 3, only 38% of English word forms are hapax, compared to 49% of German.

Language	Tokens	Forms	Hapax	Other Rare	Frequent
English	29,077,024	101,967	39,200 (38%)	35,608 (35%)	27,159 (27%)
German	27,643,792	286,330	140,826 (49%)	98,126 (34%)	47,378 (17%)

Table 3: Corpus characteristics (Schrader 2006)

Of 512 German hapax legomena examined, 68.95% are noun compounds, and 68% align with English multi-word expressions, mostly noun sequences, possibly preceded by an adjective or followed by a prepositional phrase. Correspondence of German compound length to English multi-word phrase length is high (Table 4).

English	German compound length				
	1	2	3	4	>4
1 word	59	2	1	0	3
2 words	30	119	56	15	10
3 words	2	20	15	8	10
4 words	0	1	0	0	2

Table 4: German compound length vs. English phrase length (Schrader 2006)

The English phrase length is often one more than the compound length, due to the presence of a prepositional phrase. For example, *Kongresh-vorlage*, 'submission to Congress'.

Observation: Table 4 only seems to support this observation for German noun compounds of length 1 or 2.

For longer compounds especially, a more precise analysis might lead to improved heuristics.

Approach

Instead of word-to-word mappings, Schrader mapped German compound nouns to English word sequences.

1. A German token is considered a noun compound if it is POS tagged as a noun and is at least 12 characters long.
2. English candidate phrases are generated from aligned noun sequences with optional preceding adjective and/or following prepositional phrase.
3. The length ratio is compared:
$$0 < 1 - (\text{German compound length} / \text{English phrase length})$$
4. A qualifying candidate is added to bilingual dictionary.

Results

Using this approach, 1600 additional entries were added to the bilingual dictionary. Of 53 recognized hapax compounds, 236 were translated; 248 after improvements to nominal recognition heuristics. Eleven unidentified compounds did not meet length threshold.

Of the hapax legomena added to the dictionary, 47% were translated correctly, 70% after heuristic improvements.

No baseline for comparison exists, confirmed in a personal correspondence from Bettina.

Side argument: Compositional analysis of compound nouns

Schrader argues that German compounds should not be broken into components, since the meaning of the compound may not be compositional. For instance, the German word *Personen-stand* is made up of roots literally translated as 'personal status', but the meaning of the compound is 'marital status'. However, a modern PBSMT system will

take this into account, translating the sequence as a non-compositional unit when possible, and as compositional components otherwise.

1.2.3.6. Goldwater & McClosky (2005): Aligning morphologically rich languages

Goldwater & McClosky (2005) addressed the problems of translating from a morphologically rich language (Czech) to a morphologically poor language (English).

For data, they used:

- PCEDT corpus (Čmejrek et al. 2004)
- ~21,000 sentences translated from English to Czech and morphologically annotated, for training alignment/translation model
- ~50,000 English sentences, for training language model
- 250 sentences each dev/test, translated from Czech to English by 5 translators, for evaluation

Evaluation was based on word-based (not phrase-based) MT BLEU scores.

Motivation

Czech's morphological richness leads to high data sparseness. While the numbers of rare lemmas in Czech and English are comparable, Czech has roughly twice as many rare inflected forms as English (Figure 1).

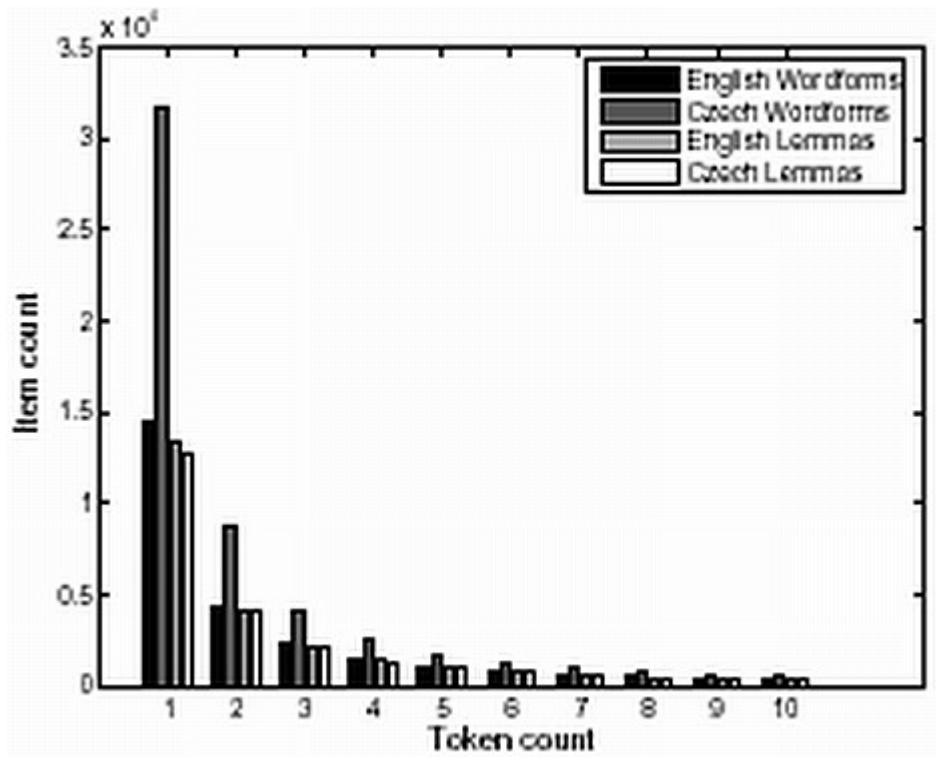


Figure 1: Rare word forms (Goldwater & McClosky 2005)

Furthermore, features represented morphologically in Czech are often represented by other means in English (Table 5).

	Czech	English
a.	plural	plural
	past tense	past tense
	person	pronouns
b.	negation	<i>not</i>
	genitive case	<i>of</i>
	instrumental case	<i>by</i> or <i>with</i>
c.	gender	none
d.	other case markings	syntax

Table 5: Morphological mappings: Czech-English (Goldwater & McClosky 2005)

Approach

A variety of approaches were compared. These are illustrated in Table 6, and explained following.

a. Original	Pro někoho by její provedení mělo smysl .
b. Lemmas	pro někdo být jeho provedení mít smysl .
c. Morpheme tokens	pro někdo být PER_3 jeho provedení mít PER_X smysl .
d. Modified lemmas	pro někdo být+PER_3 jeho provedení mít+PER_X smysl .
e. Vector	(pro) (někdo) (být PER_3) (jeho) (provedení) (mít PER_X) (smysl) (.)

Table 6: Transformations (Goldwater & McClosky 2005)

Lemmatization

Replacing each word with its associated lemma reduces data sparseness, but also loses information (09.b). This approach is expected to help when morphological information is not carried over into English (08.c,d), but may hurt in other cases.

Several different approaches to lemmatization were tried.

1. Lemmatize only certain POS.
2. Lemmatize all POS except pronouns.
3. Lemmatize only rare words.
4. Truncate to n characters.

Morpheme tokens

In this approach, words are replaced by token sequences, each token representing one morpheme (09.c). This reduces data sparseness, although not quite as much as lemmatization. This approach is expected to help most when the Czech morpheme corresponds to an English function word (08.b).

Modified lemmas

First each word is lemmatized, then tags representing morphemes of interest are concatenated to the lemma (09.d). This eliminates some variation among word forms, thus slightly reducing data sparseness. This approach is expected to help when morphological information is carried over into English (08.a).

Feature vectors

Each word is replaced by a feature vector including the lemma and tokens representing morphemes (09.e). This reduces data sparseness without loss of information, which should lead to optimum results. However, alignment is estimated using a variant of the EM algorithm, which may result in loss of accuracy.

Combined model

Based on the results of the above experiments (10) (11), they created a combined model using tokens for person and negation morphemes (08.b), and the modified lemma method for number and tense (08.a). The results are in (12).

Results

Experiments were with a word-to-word translation system. Explanation follows.

BLEU score difference of 00.9 is significant ($p < .05$).

	Dev	Test
a. word-to-word	31.1	27.0
b. truncate (k=6)	35.3	28.3
c. lemmatize all	35.5	29.9
d. except Pro	35.0	–
e. except Pro, V, N	34.6	–
f. n < 50	37.0	30.6

Table 7: BLEU scores for baseline & lemmatization

	Tokens	Mod-Lem	Vectors
a. PER	36.5	35.6	35.6
b. TEN	36.5	36.1	36.4
c. PER, TEN	35.5	36.2	35.5
d. NUM	35.4	36.7	36.1
e. CASE	35.3	34.0	33.7
f. NEG	35.7	35.6	35.3

Table 8: BLEU scores (dev set) for method & class of tag

	Dev	Test
combined model	39.0	33.3

Table 9: BLEU scores for combined model

The best lemmatization results occurred with lemmatizing low-frequency words (Table 7.f). Truncation (Table 7.b) performed better than baseline (Table 7.a), but worse than the best lemmatization.

Scores on other experiments (Table 8) were reported on the dev set only. Surprisingly, none scored better than simple lemmatization.

Adding person-agreement tokens (`PER`) helped. Examination confirmed that they did, indeed, align with English pronouns. Inexplicably, tense-marking tokens (`TEN`) helped just as much, but using both `PER` and `TEN` canceled the benefits of either.

For modified lemmas, number (`NUM`) and tense tags helped, which makes sense since these are also marked in English. Case tags hurt, which also makes sense, since they increase data sparseness without providing any information marked in English.

Disappointingly, feature vectors never significantly outperformed morpheme tokens.

The combined model performed best of all, successfully combining the advantages of each method.

Conclusions

Morphological analysis helps, through the reduction of data sparseness. The greatest improvement comes from changing the source text to reflect the morphological characteristics of the target text.

1.2.3.7. Lee (2004): Aligning templatic languages

Young-Suk Lee (2004) addressed problems of translating Arabic, a templatic language, with English. He observes that in Arabic→English translation, one Arabic word frequently aligns with multiple English words, owing to functional affixes in Arabic, e.g.: *llmEArDp* ↔ "*of the opposition*". This one-to-many alignment poses technical challenges to PBSMT systems.

Lee's approach takes several steps:

1. POS tag Arabic and English parallel text.
2. Segment Arabic words into prefix(es)-stem-suffix(es) as distinct tokens.
3. Align segmented Arabic text with English text.
4. Determine translation probability of each Arabic stem/affix and POS into English POS.
5. Evaluate each Arabic affix.
 - (a) If the affix robustly translates to an English POS, keep it as a token.
 - (b) If the most common translation for the affix is NULL, delete it.
 - (c) Else, merge the affix back into the stem.
6. [*PBSMT experiments only*] Manually delete multiple definite determiners in a single Arabic phrase.

Lee evaluates the effects of his approach on BLEU score for corpora ranging from 3.5K to 3.3M words, using either IBM Model 1 (a WBSMT system) or phrase-based alignment (Table 10).

# sentences	Model 1		PBSMT	
	base	morph	base	morph
3.5K	.10	.25	.17	.24
35K	.14	.29	.24	.29
350K	.18	.31	.32	.36
3.3M	.18	.32	.36	.39

Table 10: BLEU-score evaluation of delete/merge (Lee 2004)

For Model 1, BLEU scores are vastly improved using Lee's method, roughly doubled.

For PBSMT, the improvements are roughly equivalent to an order of magnitude of data.

1.2.3.8. Habash & Rambow (2005): Arabic morphological complexity

Habash & Rambow (2005) also looked at Arabic→English translation. They noted that English has 50 morphological classes, while Arabic has roughly 333,000 in theory, 2200 attested.

1.2.3.9. Isbihani et al. (2006): Segmenting Arabic

Isbihani et al. (2006) compared the impact on MT of various methods for segmenting Arabic source text. They found that the best results came from an unsupervised method using a finite-state automaton augmented with a memory of word properties.

1.2.4. Related work: Crosslingual word sense disambiguation

Crosslingual word sense disambiguation (CWSD) uses words of the target language as the sense inventory for words of the source language. For instance, if the source language word *bank* can translate to either *banque* or *encaisser* in the target language, then [*banque*, *encaisser*] are considered to be possible senses for *bank*. This nicely solves theoretical questions in WSD such as what level of granularity to use for word senses. If it leads to a different translation, it's a different sense; if it doesn't, it's not.

1.2.4.1. Vickrey et al. (2005)

Vickrey et al. (2005) describe CWSD results for data taken from the Europarl fr-en parallel corpus (Koehn 2005). Their baseline always guesses the most common sense for a given word, yielding an F-score of 51.1%. Their logistic regression model yields an F-score of 62.0%.

They also describe an experiment with filling in a missing ambiguous word in a target sentence. This was intended to prove the potential usefulness of WSD in machine translation.

1.2.5. Related work: WSD & machine translation

Attempts to use WSD as a preprocessor for MT have generally not produced encouraging results (Och 1999; Carpuat & Wu 2005; Cabezas & Resnik 2005). One explanation is that WSD artificially increases data sparseness, without focusing on senses that are

relevant to translation (Yarowsky & Florian 2002). This is one motive for exploring crosslingual WSD, using target language words as the sense inventory for source language words (Resnik & Yarowsky 2000). In fact, one of the exercises in Senseval-3 was crosslingual (Mihalcea & Edmonds, ed. 2004).

Until recently, most research in WSD involved only annotated text. Some have begun looking into semi-supervised learning, augmenting a small amount of annotated text with a large amount of unannotated text, with optimistic results (Yarowsky 1995; Pham et al. 2005). Such an approach would be particularly useful when translating from a high-resource language into a low-resource language, since the amount of parallel text is likely to be small, while the amount of unannotated text in the source language is virtually without bound.

1.2.5.1. Carpuat & Wu (2005): Constrain MT using WSD

Carpuat & Wu (2005) used WSD to constrain possible translations, in contrast to our approach of providing probabilistic lexical selection recommendations. They concluded, "Even state-of-the-art WSD does not help BLEU score."

1.2.5.2. Cabezas & Resnik (2005): Improve MT using WSD

Cabezas & Resnik (2005) attempted to use crosslingual WSD to improve Spanish→English MT. They found no significant effect. They performed no lemmatization or morphological analysis; nor did they identify content words.

1.2.5.3. Pham et al. (2005)

Pham et al. (2005) used *sgt_light* (Joachims 2003) on tasks similar to ours, with positive results.

Pham et al. published their results just as I was beginning my research, after I had completed a literature review. I did not encounter their paper until after I had written up my first round of results. Hence the relative lack of prominence of their work in my write-up.

1.2.6. Related work: Phrase-based statistical machine translation (PBSMT)

1.2.6.1. Chiang (2005): Hiero

At the time of these experiments, the University of Maryland boasted the state-of-the-art PBSMT system, Hiero (Chiang 2005).

Previous PBSMT systems (e.g. Koehn 2004; Och & Ney 2004) generated translation candidates at two levels, word-level and phrase-level, where a *word* is any contiguous sequence of characters delimited by white space (i.e. a token), and a *phrase* is any contiguous sequence of words. The primary innovation of Hiero is to generate translation candidates from a hierarchy of phrases of arbitrary depth (*i.e. phrases made up of phrases*).

One effect of PBSMT in general, and hierarchical PBSMT in particular, is to mitigate the impact of overzealous tokenization. Two "words" which should not have been separated

are simply grouped together by the system into functional "phrases". Thus, we expect to benefit by using the most aggressive morphological analysis available.

Caveat: Both Pharaoh and Hiero use a configurable maximum number of "words" per "phrase". Breaking words into their component morphemes will increase the length of any given "phrase". The phrase-length limit will therefore need to be increased, which is likely to increase processing time.

Hiero, like Pharaoh before it (Koehn 2004), provides a probabilistic lexical selection mechanism. The preprocessor can specify one or more proposed translations for a word or phrase, each with a feature label (e.g. "WSD") and a specified probability or cost. Feature weights are then optimized based on a dev set.

1.2.7. Related work: Machine learning

1.2.7.1. Joachims (2003): Spectral Graph Transducer (SGT)

Joachims (2003) introduced the Spectral Graph Transducer (SGT) technology to perform semi-supervised machine learning using a small amount of annotated training data and a large amount of observed test data, so-called *transductive learning*. A trivial adaptation uses small amounts of test and training data and a large amount of unannotated data.

Joachims points out that SGT subsumes various cotraining tasks (Yarowsky 1995; Blum & Mitchell 1998) as a special case. More specifically, SGT handles cotraining applications where the feature set for each view remains static across iterations.

In brief, SGT constructs a graph with examples as vertices and edge weights indicating similarity, and finds an approximate minimal cut between positive examples and negative examples using a spectral graph method.

Some advantages of SGT are that it is extremely fast, even with large numbers of features and examples; it can be trained independently of the classification values (i.e. +1 | -1), allowing multiple models of the same data to share graph files; and it can easily combine multiple graphs based on the same examples but different models, thus implementing various cotraining scenarios with a minimum of effort.

1.2.8. Related work: Corpus linguistics

1.2.8.1. Babych & Hartley (2003): S-score

In a paper on named entity recognition in MT, Babych & Hartley (2003) introduced the *s-score* heuristic for identifying content words in a large corpus.

2. TASK I: UNSUPERVISED MORPHOLOGICAL ANALYSIS

2.1.1. Task formulation

2.1.1.1. Hypothesis

These experiments were designed to test the hypothesis:

We can design a system for unsupervised morphological analysis that applies to any morphologically complex language, and has accuracy comparable (not necessarily superior) to language-specific systems.

2.1.1.2. Objective

Naturally, the first step in designing a morphological analysis system is to answer the question: What is a morpheme?

We don't.

Or, rather, we adopt a utilitarian approach: A morpheme is whatever can be analyzed constructively as a morpheme for our intended purpose. If our purpose is word sense disambiguation, a morpheme is any substring of a word such that treating it as a morpheme improves the accuracy of word sense disambiguation.

As an example, take the English word *implementation*. An expert might say this is two morphemes: *implement* + *-ation*. Our system may infer, instead, that it consists of five

morphemes: *im-* + *ple* + *-ment* + *-a* + *-tion*. (Note that *im-*, *-ment*, and *-tion* are all frequently attested.) If it turns out this analysis works well for our evaluation metric, we don't dwell unduly over the implausibility of *ple* as a stem.

Our approach may therefore benefit considerably from optimization based on our choice of evaluation metric(s).

2.1.1.3. Languages of interest

In selecting languages for experimentation, we wanted to cover as wide a range of morphological types as possible, including:

- agglutinative – e.g. German, Turkish, Inuktitut
- fusional – e.g. English, Spanish, Finnish
- suppletive – e.g. Czech
- templatic – e.g. Arabic

Listed in order of expected difficulty. Fully analytic languages are a moot point, since they have no morphology to speak of.

Disclaimers: It's been disputed whether German is truly agglutinative, as opposed to fusional. It appears to function so for our purposes. English is claimed by some to be virtually analytic.

For our initial experiments, we decided to begin with simpler languages (i.e. agglutinative and fusional), and move to more difficult languages (i.e. suppletive and

templatic) later on. We were also constrained by pragmatic factors, e.g. availability of data, particularly parallel data (for downstream applications); readability to a monolingual English speaker (me); and demand within the broader research community.

We decided to begin with a selection of Europarl languages, using data available from the parallel corpus (Koehn 2005). Specifically, we examined:

- German (agglutinative, otherwise morphologically poor)
- English (fusional, morphologically poor)
- Spanish, French (fusional, morphologically moderate)
- Finnish (fusional, morphologically rich)

By fortunate coincidence, WMT06 later used the same data source.

2.1.1.4. Preprocessing methods

Several simple preprocessing heuristics and algorithms were evaluated. These included:

- Truncate each word to n characters, $4 \leq n \leq 7$.
- Strip off inflectional suffixes known to be common, such as *-e* and *-s* in French.
- Strip out diacritics.

Our primary algorithm was based on (Koehn & Knight 2003) as a starting point. We chose their approach on the premise that it could be equally applicable to most other languages (agglutinative, fusional, or suppletive).

A key difference between German and fusional (particularly Romance) languages quickly became apparent. A German noun compound is built from an open class of unbound roots (nouns), each independently attested; and a small, closed class of bound fillers (e.g. *-a-*, *-s*).

Disclaimer: This description is based upon anecdotal observation, and may not be generally accurate.

In contrast, a word in a fusional language is built from an open class of (mostly) unbound roots, most of which are independently attested; a large (although probably closed) class of bound affixes; and a virtually endless array of morphosyntactic variations

One apparent exception in English is: *inept* ↔ *in-* + *ept* (?), where *in-* is the frequently attested negative prefix, but *ept* is not attested in any other context that I am aware of.

Presumably it was a word in some earlier incarnation of the language, perhaps a variation of apt.

As another example, note the apparent morphological pattern in: *ascend*, *descend*, *transcend*.

The end result is that morphological analysis in German is a fairly straightforward statistical analysis of observable events (i.e. known roots), while English analysis involves statistical analysis of events which are themselves hidden (i.e. proposed morphemes). This description suggests EM, which, indeed, is similar to the iterative approach we ended up using.

Morphological analysis algorithm

Our algorithm is (roughly) as follows:

1. Enumerate all words (i.e. word forms) from a corpus, with the number of occurrences of each word.
2. Posit initial selection of morphemes.
 - (a) Store the counts of all attested words.
 - (b) Add counts for all likely affixes, taken as the first or last n characters, $1 \leq n \leq 3$.
3. Morphologically analyze each word in the corpus.
 - (a) Give the unsplit word a score based on $\log(\text{count}+1)$, modified for various configurable heuristics.
 - (b) Enumerate all possible binary splits.
 - (c) For each possible split, determine its score as the weighted mean of the scores of each half, again modified by heuristics. The weight reflects the number of morphemes posited for the substring.
 - (d) Choose the 1-best analysis as the hypothesized split for that word.
4. Do it again.
 - (a) Use the hypothesized splits to generate new counts for each (newly) attested morpheme.
 - (b) Run (3) again using these new counts.
 - (c) Repeat until convergence, or maximum n iterations. (We used $n=20$.)

We experimented with various options, including:

- Posit initial selection of morphemes.
 - Exclude words without any vowels.
 - Exclude words below n characters long, $1 \leq n \leq 4$.
 - Use content words only, by excluding words below a configurable s-score threshold (Babych & Hartley 2003).
- Morphologically analyze each word in the corpus.
 - Bias against an affix without a vowel.
 - Bias against an affix with only one letter.
 - Bias against novel affix (count=0).

Ideally, we should set up a procedure to optimize these configuration options (and others) based on development data and evaluation metric.

2.1.2. Evaluation

2.1.2.1. Complexity

Morphological analysis can be viewed as an exercise in the reduction of *complexity* without loss of *information*. Stand-alone methods of evaluation are designed to reflect this goal.

Rare events

Rare events are a bane of statistical applications, as indicated by (among others) Schrader (2006) for German, and Goldwater & McClosky (2005) for Czech. Prima facia, a statistical system has a 0% chance of processing hapax legomena correctly, since they never occur in both test and training data. While most languages are not as bad as German or Czech, Zipf's law applies. Morphological analysis can have a potentially profound impact by reducing these intractable or nearly intractable words into recognizable component morphemes.

The problem may not be as bad as it seems at first. Even if half the forms are hapax (as for German or Czech), the percentage of singleton tokens will be considerably lower, typically less than one percent of all tokens (14). On the other hand, these hapax tokens are likely to be content words, critical to the adequacy of translation (or whatever your end purpose), whereas the highest frequency forms are most likely to be function words.

In other words, rare events are far more critical to correct translation than is captured by their relative impact on BLEU score or similar automated measures.

We performed analysis similar to that in Schrader (2006) for our languages of interest, before and after breaking words into component morphemes. For the purpose of analysis, we considered "rare" any form that occurs 10 or fewer times in the corpus (Table 11).

		Total		Frequency 1 (<i>hapax</i>)		Frequency 1-10			
		# Forms	# Tokens	# Tokens	% Forms	% Tokens	# Tokens	% Forms	% Tokens
<i>German</i>	Words	170,402	13,233,854	81,052	47.57%	0.613%	140,464	82.43%	2.377%
	Morphemes	30,898	13,348,182	9,836	31.83%	0.074%	19,507	63.13%	0.385%
<i>English</i>	Words	34,827	10,648,289	9,361	26.88%	0.088%	21,350	61.82%	0.587%
	Morphemes	18,955	10,659,570	3,991	21.06%	0.038%	9,817	51.79%	0.285%
<i>Spanish</i>	Words	77,526	12,495,786	25,772	33.24%	0.201%	54,908	70.83%	1.182%
	Morphemes	32,407	12,530,435	8,102	25.00%	0.065%	18,334	56.57%	0.425%
<i>French</i>	Words	55,985	12,036,764	16,404	29.30%	0.136%	37,045	66.17%	0.875%
	Morphemes	25,880	12,058,591	5,756	22.24%	0.048%	13,524	52.26%	0.334%
<i>Finnish</i>	Words	301,597	7,835,581	155,994	51.72%	1.990%	259,890	86.17%	7.102%
	Morphemes	38,019	8,085,040	12,238	32.19%	0.151%	24,095	63.38%	0.788%

Table 11: Rare events, by language, with and without morphological analysis

As expected, the results are dramatic for German, a compounding language, where the number of distinct forms drops 82% from 170,402 to 30,898, and the number of rare tokens drops 86% from 140,464 to 19,507. The results are even more dramatic for Finnish, a morphologically rich language, where the number of distinct forms drops 87% from 301,597 to 38,019, and the number of rare tokens drops 91% from 259,890 to 24,095. The reduction is also considerable for other languages. Even English, a morphologically poor language, sees reductions of around 50%.

Note that the number of tokens necessarily goes up as words are split into component morphemes. However, the amount of increase is not as significant as one might expect. Presumably, this reflects a characteristic of language that most common words are morphologically simple. It also confirms that we are making informed decisions about splitting morphemes, doing so only where it reduces complexity.

Frankly, the increase in number of tokens seems ridiculously low. One would expect it to be at least the same magnitude as the decrease in forms, but this is not always the case.

Contrast the naive method of treating every character as a morpheme (not shown). We would predict the number of forms to drop down to the size of the alphabet, but the number of tokens to jump to the number of characters in the corpus.

Or contrast another naive method, truncating every word to n characters (also not shown). We would expect this to dramatically reduce the number of distinct forms, and virtually eliminate rare events, while keeping the number of tokens constant. This demonstrates the theoretical limitation of this evaluation metric, in that it does not account for the loss of information involved in truncation. It also provides a possible indication why truncation performs so well in many applications.

Schrader (2006) used a totally different approach, both to simplification and evaluation, making it difficult to compare results.

Complexity vs. information

Various metrics have been used to measure complexity vs. information in morphological analysis, e.g. *saturation* and *informativeness* (Baldwin 2005).

2.1.2.2. Comparison against a standard

One can also evaluate against an established standard. Possible candidates include gold-standard data with (manual) morphological annotations (if any can be found); automated

output of an existing (presumably rule-based) system; and CELEX (Baayen et al. 1995) – a popular choice in the literature (e.g. Freitag 2005, Wicentowski 2002).

We do not expect our method to outperform language-specific rule-based systems, although it would be interesting to see how closely they compare.

2.1.2.3. Downstream applications

Finally, one can evaluate the effects of morphological analysis on downstream applications. We looked at word sense disambiguation and machine translation (both described below), although results are incomplete.

3. TASK II: CROSSLINGUAL WORD SENSE DISAMBIGUATION

These sets of experiments endeavored to improve the quality of crosslingual word sense disambiguation (CWSD) using a variety of approaches, including:

1. Task II.A: Incorporate a moderate-to-large quantity of unannotated source language data.
2. Task II.B: Incorporate document-level features.
3. Task II.C: Incorporate unsupervised morphological analysis of the source language data.
4. (*future work*) Incorporate unsupervised morphological analysis of the target language data.

3.1. Task II.A: WSD using unlabeled data

This set of experiments evaluated the effects on WSD results of supplementing annotated training data with unlabeled data, a semi-supervised approach known as *transductive learning*. The theory is that patterns in the unlabeled data can help elucidate patterns in the labeled data.

Historically, this was my first set of original experiments, in fall 2005. The results provided part of the impetus to pursue morphological analysis.

3.1.1. Task formulation

3.1.1.1. Hypothesis

These experiments were designed to test the hypothesis:

Accuracy of word sense disambiguation can be improved by providing unannotated data in addition to annotated training data.

3.1.1.2. Software

For the core engine, we employed two machine learning packages, *svm_light* and *sgt_light*, both made publicly available by Thorsten Joachims.

svm_light is an implementation of support vector machines (Joachims 1999). Although *svm_light* includes a transductive learning option, it runs too slowly on large data sets to be of practical use.

Update: Joachims has since released a new version of svm_light which is supposed to perform much better on larger data sets.

sgt_light is an implementation of spectral graph transducers (Joachims 2003), implementing binary (yes-no) classification only. We therefore trained a set of binary models for each word form, one model for each possible sense. We chose the best sense based on the highest-confidence model.

With *sgt_light*, the test data *must* be present when the model is built, so it is not possible to run experiments using training data only.

To align text, we used Pharaoh (Koehn 2004) wrapping GIZA++ (Och 2003).

3.1.1.3. Data

For experimental data, we used Senseval WSD evaluation sets, and also data constructed from parallel corpora.

interest & line

The initial set of experiments used the Senseval-2 data (*available on* Ted Pedersen's website) for performing (monolingual) WSD on the words *interest* and *line*. Each set was randomly divided into 75 test examples, 150 labeled training examples, and the remainder unlabeled examples, 2143 for *interest* and 3921 for *line*.

This was the reality-check stage. Since all the data came preprocessed from the same source, there was (presumably) no noise or comparability issue.

Spanish Lexical Sample

The next set of experiments used the (monolingual) SpanishLS data from Senseval-3.

The data came preprocessed into 4195 test examples, 8430 labeled training examples, and 61252 unlabeled examples, covering a total of 46 lexical roots, ranging from 2 to 8 senses per root.

Europarl fr-en

To anticipate an MT scenario, we constructed a CWSD exercise from parallel text. We chose the Europarl fr-en corpus (Koehn 2002), since it is large enough to produce many words and examples, and we have at least a fighting chance of interpreting both the source and target language texts.

Starting with the intersected fr-en and en-fr alignments, we generated a list of all the French content words, as indicated by s-score, and their aligned English words. Any alignment that occurred fewer than 6 times was lumped into UNK. Potential words of interest were identified with at least 500 examples, between 3 and 7 senses, no more than 5% UNK alignments, and no more than 65% of examples for any one sense. A visual scan filtered out remaining candidates whose different alignments represented only affixation variants. (*This predated our work on morphological analysis.*) That left a total of 42 French words and their English alignments. For each of these 42 words, the examples were randomly divided into 50 test examples, 50 dev examples, 150 labeled training examples, and the remainder (at least 250) unlabeled examples.

Europarl en-fr

We again processed the parallel text from the Europarl fr-en experiment, only this time with English as the source language and French as the target language. The resulting data set contained 33 English words and their French alignments.

3.1.1.4. Feature Analysis

With each data set, we explored various methods of analyzing the data into features, optimizing for each set. Not all configuration options are described here. Comparable experiments are configured identically except as indicated.

Preprocessing

Preprocessing options included the following.

1. *punctuation*: keep (as distinct tokens) or strip out
2. *case*: lowercase or keep mixed
3. *diacritics*: keep or convert to standard characters
4. *digits*: keep, replace with '#' character, or strip out all words containing digits

Feature Selection

Every experiment used at least local context and broad context options for feature selection. Later experiments also included other feature sets.

1. *local context*: head token and tokens within a 3-word window to either side of the word being examined
2. *broad context*: all tokens occurring anywhere in the example
3. *bigrams*: contiguous word pairs taken from local context
4. *document*: the label of the source document, intended to emulate the one-sense-per-document heuristic of (Yarowsky 1995)

Following (Pham et al. 2005), we experimented with two alternatives for using multiple feature sets.

1. lump all features together into one big model
2. take each feature set as a different view, and cotrain

Counts

We explored different options for counting features, following (Cabezas & Resnik 2005).

1. *counts*: full counts, log counts, or simple boolean (present or not present)
2. *IDF*: multiplier or no multiplier

Inverse Document Frequency (IDF) is essentially a hack that gives the system an extra hint as to which features are likely to be most significant. Various heuristics are possible, but they all follow the basic pattern of penalizing features that occur in examples from many different classes, and rewarding features that occur in only one or a few classes. We tried several heuristics, but do not discuss the results here.

It turned out that using a simple boolean for counts always produced the best results, but the optimal setting for IDF varied by data set.

3.1.1.5. Evaluation

F-score

We calculated the standard F-score for each experiment.

Baseline

We compared against two baselines:

1. For every occurrence, always predict the most frequently occurring sense.
2. Classify senses using *svm_light* with training data only (i.e. no unlabeled data).

The first (naive) baseline was just a sanity check, since the second was always higher, as expected. The second baseline provided the target to beat.

However, quirks of *svm_light* and *sgt_light* led to a practical problem. *svm_light* could run on small data sets (e.g. training data only), but not on large data sets. In contrast, *sgt_light* could run on large data sets, but not on training data only. We therefore could not use a single tool to get the direct head-to-head comparison that would most clearly demonstrate our hypothesis.

Filtering

Later experiments included filtering options, trading recall for precision.

1. *confidence*: return only results above a confidence threshold (typically 50%)
2. *dev*: return results only for words which scored above an F-score threshold (typically 50%) on a dev set

The results with filtering are not presented in this paper. It generally had little effect on F-score.

3.1.2. Results: WSD using unlabeled data

F-score results are in Table 12, with explanations following.

		Data available		
		training	+ test	+ unlabeled
interest	baseline	50.7		
	svm_light	69.3	72.0	81.3
	sgt_light		76.0	85.3
	sgt-co		80.0	85.3
line	baseline	60.0		
	svm_light	72.0	69.3	81.3
	sgt_light		68.0	80.0
	sgt-co		69.3	77.3
Spanish LS	baseline	67.7		
	svm_light	83.2	85.1	–
	sgt_light		84.6	86.6
	sgt-co		85.1	85.7
Europarl fr-en	baseline	56.7		
	svm_light	70.7	66.0	–
	sgt_light		65.2	68.9
	sgt-co		65.5	69.4
Europarl en-fr	baseline	51.3		
	svm_light	64.7	61.1	–
	sgt_light		60.4	62.8
	sgt-co		61.2	63.5

Table 12: F-score WSD results

The columns are:

- training only – system trained with labeled training data only
- + test – system trained with labeled training data and unlabeled test data available
- + unlabeled – system trained with training, test, and additional unlabeled data available

The rows are:

- baseline – always choose most common sense
- svm_light – *svm_light* trained on indicated data
- sgt_light – *sgt_light* with all features lumped together into one model
- sgt-co – *sgt_light* with local and broad context features in separate cotrained models

Empty cells denote impossible experiments. Cells with a dash '-' denote theoretically possible experiments that were beyond the capability of the software.

Experiments that indicate locally negative results (i.e. lower score than the cell to its immediate left) are in **red bold**. Experiments that indicate locally positive results (i.e. higher score than the cell to its immediate left) are in **blue bold italic**.

interest & line

Providing additional unlabeled data produced considerably better results than train & test data alone, thus supporting our experimental hypothesis. One surprising result was that *svm_light*'s accuracy on *line* data (and, later, the Europarl data sets) actually degraded when the test examples were made available.

My best guess is that the explanation lies in the implementation of the svm_light software. Including unlabeled data (e.g. test data) invokes the transductive learning option, which turns off some optimization routines. I suspect that these routines not only speed up performance, but actually enhance the accuracy. However, this hypothesis is not supported by the results reported in (Joachims 2003). Note that adding the rest of the unlabeled data did, in fact, improve the performance of svm_light over training data alone.

It made little difference whether local and broad context were combined into one model, or used to generate two separate models which were then cotrained. This contrasts with the findings in (Pham et al. 2005), which showed better results with cotrained models.

Our results for train & test data are comparable with those reported in (Pham et al. 2005), while our results with unlabeled data are much higher. Although unclear from their paper, it appears that (Pham et al. 2005) did not make use of unlabeled data other than the test data.

To the best of my knowledge, our results were higher than any reported in any research literature.

Spanish Lexical Sample

The *svm_light* baseline was much higher than with interest & line. Adding unlabeled data marginally improved the results. The results are consistent with our experimental hypothesis, but do not strongly support it.

Note that our system outperformed the best system entered in Senseval-3 (Marquez et al. 2004). However, this is not a fair comparison, since we optimized directly on the test data. It would be interesting to run the experiments again with proper controls.

Europarl

With both Europarl fr-en and en-fr, unlabeled data noticeably improved accuracy over just train & test data. However, even with all unlabeled data, the accuracy of *sgt_light* was less than the accuracy of the *svm_light* baseline. This highlights the practical problem of not being able to run a whole test suite using a single tool.

It should now be possible to run the entire set of experiments using just the latest version of svm_light, allowing direct comparison of results.

Error analysis suggested that one reason we didn't see stronger results was data sparseness introduced by morphological complexity. For instance, the English word *student* is considered to have four senses [*étudiant, étudiante, étudiants, étudiantes*], even though a human observer can easily see there is only one. A morphologically rich target language creates data sparseness by exploding the sense inventory, as in this example. A

morphologically rich source language creates data sparseness by exploding the lexicon, e.g. storing [*étudiant, étudiante, étudiants, étudiantes*] all as distinct lexical entries.

A traditional statistical approach would address the problem by throwing more data at it. This might be feasible with resource-rich languages such as French and English, but will not be viable when working with less studied languages. Rather, this suggests the need for morphological analysis, at least on the side of the morphologically rich language.

See: WSD using morphological analysis.

3.2. Task II.B: WSD using document-level features

3.2.1. Task formulation

Yarowsky (1995) demonstrated that document-level information, even comparatively primitive, can significantly enhance the quality of unsupervised WSD, to levels comparable with supervised methods. However, data provided for Senseval (now Semeval) conferences have always involved individual sentences only. New data sets need to be constructed, including document-level context, to allow experimentation with document-level features.

3.2.1.1. Hypothesis

These experiments were designed to test the hypothesis:

Accuracy of word sense disambiguation can be improved by utilizing document-level features, in addition to standard sentence-level features.

3.2.2. Results

Incomplete. *See:* Future work related to WSD with document-level features.

3.3. Task II.C: WSD using morphological analysis

3.3.1. Task formulation

3.3.1.1. Hypothesis

These experiments were designed to test the hypothesis:

Accuracy of word sense disambiguation can be improved by performing (unsupervised) morphological analysis on the text.

3.3.1.2. Motivation

Early CWSD experiments involving English and French were hampered by data sparseness, caused, in large part, by morphological complexity.

3.3.1.3. Data

For these experiments, data sets were constructed from the Europarl parallel corpus, as described above under Task II.A. Languages of interest are as described under Task I.

3.3.2. Results

3.3.2.1. Results: Naive morphology

As a proof of concept, we first ran experiments using naive morphological preprocessing, such as is typical in other applications. Since our focus was source language morphology,

we used our morphologically simplest language (English) as the target language. We specifically used:

- Language pairs
 - German→English (de-en)
 - Spanish→English (es-en)
 - French→English (fr-en)
 - Finnish→English (fi-en)
- Preprocessing options
 - none : baseline, no preprocessing
 - trunc N : truncate to N characters
 - lc : lowercase all
 - diac : eliminate diacritics
 - suffix : strip common inflectional endings (-e and -s)
- Evaluation
 - F-score on dev set
- Statistics
 - Forms : number of distinct word forms being disambiguated
 - Senses : total number of distinct senses (i.e. translations) for all forms
 - Examples : total number of training examples for all forms
 - S/F : average number of senses per form
 - E/F : average number of examples per form
 - E/S : average number of examples per sense

Results are shown in Table 13, sorted by language in decreasing order of F-score.

Languages	Process	F-score	Forms	Senses	Examples	S/F	E/F	E/S
de-en	trunc6	0.575	99	1505	127071	15.20	1283.55	84.43
	trunc5	0.574	96	1961	155221	20.43	1616.89	79.15
	diac	0.559	101	598	45048	5.92	446.02	75.33
	none	0.558	101	587	44330	5.81	438.91	75.52
	lc	0.558	101	587	44330	5.81	438.91	75.52
es-en	trunc6	0.614	99	1006	101280	10.16	1023.03	100.68
	trunc5	0.608	98	1490	136166	15.20	1389.45	91.39
	diac	0.604	104	618	50637	5.94	486.89	81.94
	none	0.598	104	610	46811	5.87	450.11	76.74
	lc	0.598	104	610	46811	5.87	450.11	76.74
fr-en	strip	0.575	97	825	93046	8.51	959.24	112.78
	trunc6	0.598	75	930	79461	12.40	1059.48	85.44
	trunc5	0.593	74	1338	111140	18.08	1501.89	83.06
	diac	0.586	80	485	36139	6.06	451.74	74.51
	none	0.584	80	478	35857	5.98	448.21	75.01
fi-en	lc	0.583	80	478	35857	5.98	448.21	75.01
	strip	0.568	77	650	66794	8.44	867.45	102.76
	trunc6	0.560	60	522	37618	8.70	626.97	72.07
	trunc5	0.557	58	755	55885	13.02	963.53	74.02
	none	0.539	63	347	25411	5.51	403.35	73.23
fi-en	lc	0.539	63	347	25411	5.51	403.35	73.23
	diac	0.538	63	347	25411	5.51	403.35	73.23

Table 13: Naive morphology WSD results

The test set is consistent across all experiments in a given language, but the training set intentionally is not.

For example, *aktionsplan* might occur 500 times in the text. After removing 50 dev examples, 50 test examples, and 150 training examples, that leaves 250 unlabeled examples. Now let's say *aktions* occurs 2500 times in the text. When we run the *trunc6* experiment, all training and unlabeled examples are grouped together for both words, along with all other words beginning *aktion-*. The example data set for the truncated form of *aktionsplan* is therefore considerably higher than for the original form. This also

introduces considerable noise, as aktionsplan might now be mapped to any word which is a translation for any word beginning aktion-.

Of these naive heuristics, only truncation has a substantial positive effect, this despite the considerable increase in noise (i.e. number of possible senses). This is therefore the target to beat with any more sophisticated approach.

Lowercasing and removing diacritics have negligible effect.

Stripping common inflectional endings has a substantial negative effect. This suggests that those endings carry more semantic weight than we may have thought.

If we throw out *strip* experiments as an outlier, there is a (weak) correlation between F-score and E/S, the average number of examples per sense. In other words, if we increase the amount of available data proportionally more than we increase the amount of noise, we win.

These experiments support our hypothesis that morphological analysis can help WSD. Now the goal is to outperform simple truncation.

3.3.2.2. Results: Morphological analysis on Europarl data

A next step would be to run CWSD on the Europarl data sets in (17).

3.3.2.3. Results: Classifier combination

As noted above, Vickrey et al. (2005) set up a Europarl fr-en CWSD experiment in which they achieved 62.0% F-score, compared to a 51.1% baseline (18). They graciously sent us their data so we could compare results. However, a direct comparison is uninformative, since they used a language-specific POS tagger, and we did not.

It would have been quite informative if we had beat them anyway, but we didn't. We could rerun the experiment using a POS tagger to get a fair comparison.

Instead, we used this set of experiments to illustrate the usefulness of another technique: classifier combination.

We ran experiments on a dev set using the following algorithms:

1. baseline – always choose the most common sense
2. unmod – WSD on unmodified data
3. trunc5 – WSD with all words truncated to 5 characters
4. trunc6 – WSD with all words truncated to 6 characters
5. morph – WSD using full unsupervised morphological analysis

Then, for each word, we chose whichever classifier had the highest precision on the dev set, and ran that on the test set. As expected, this 1-best system outperformed any of the individual classifiers (Table 14).

	Precision	# Best
<i>Vickrey baseline</i>	52.6	
<i>Vickrey best</i>	62.0	
<i>baseline</i>	52.9	595
<i>unmod</i>	55.6	540
<i>trunc5</i>	50.0	214
<i>trunc6</i>	51.5	228
<i>morph</i>	54.7	294
<i>1-best</i>	57.7	

Table 14: WSD results on Vickrey English data

The # **Best** column shows the number of base words for which each individual system proved to be the most precise.

The baseline was the best performer on a plurality of words (595/1871). That is, on any of those 595 words, using our classifier (with any options) produced worse results than always choosing the most common sense. This is an apparent mandate for using some sort of filtering or classifier combination, so that we only use our system when it is likely to help.

Of the systems using our classifier, using no morphological analysis produced the best results on a large plurality of words (540/1276), and the best overall results (55.6%). This again mandates some sort of informed classifier combination.

Of the systems using morphological analysis, our full-blown analyzer performed better than truncation on a plurality of words (294/736), and overall (54.7%). This confirms that our techniques can improve upon naive truncation in many cases.

As expected, the 1-best combined classifier performed better than the best individual classifier (57.7% vs. 55.6%). This confirms the usefulness of classifier combination.

Eyeballing a word-by-word breakdown of results (not shown) reveals that ultra-high-frequency words tend to get the best results with either the baseline or unmodified WSD. This is not surprising, given that morphological analysis is expected to help most with sparse data. It also explains the high overall scores of the simple systems.

Beyond this, it is not immediately clear how results are correlated. It's not even clear that the words where the morphological system performs best are those with the most morphological variation. For instance, *love* is one of these words, even though no other morphologically similar words occur in the corpus.

Other methods of classifier combination are also possible, of course, and should be examined. *See:* Other future work related to WSD.

4. TASK III: MACHINE TRANSLATION USING MORPHOLOGICAL ANALYSIS AND/OR WORD SENSE DISAMBIGUATION

4.1. Task III.A: Machine translation using morphological analysis

4.1.1. Task formulation

4.1.1.1. Hypothesis

These experiments were designed to test the hypothesis:

Accuracy of machine translation can be improved by performing (unsupervised) morphological analysis on the source text.

4.1.1.2. Motivation

Previous related work suggests strong potential for using morphological analysis to improve the quality of machine translation.

Data sparseness is a real problem, even with large corpora. Half of all word forms in German (agglutinative) are hapax (Schrader 2006). There are twice as many rare word forms in Czech (morphologically rich) vs. English (morphologically poor) (Goldwater & McClosky 2005). Arabic (templatic) has thousands of morphological classes (Habash & Rambow 2005).

Morphological analysis improves MT results in some cases, up to the equivalent of an order of magnitude of data (Nießen & Ney 2004, Lee 2004). The most significant improvements are from changes which take the structure of the target language into consideration (Goldwater & McClosky 2005).

PBSMT overcomes over-eager tokenization, favoring greedy approaches (Koehn & Knight 2003).

Most previous research involved techniques highly customized for a specific language pair. We planned to demonstrate utility across several language pairs, representing various language types.

4.1.2. Results

Incomplete. *See:* Future work related to MT and morphological analysis.

4.2. Task III.B: Machine translation using word sense disambiguation

4.2.1. Task formulation

4.2.1.1. Hypothesis

These experiments were designed to test the hypothesis:

Accuracy of machine translation can be improved by performing crosslingual word sense disambiguation on the source text.

4.2.1.2. Motivation

Previous related work has been less than encouraging when it comes to using CWSD to improve downstream machine translation. However, we suspect this reflects limitations in the approaches tried, rather than in the concept, itself. *This suspicion was confirmed by (Pham et. al. 2005).*

For CWSD to have a significant effect, it would presumably need to consider factors not already taken into account by the MT system. In our case, this includes broad context (i.e. document-level) features; and large quantities of unannotated source language text.

4.2.1.3. Platform

The University of Maryland owns the MT system, Hiero. This was the obvious choice of platform for our experiments. In particular, Hiero offers an option for specifying

translation candidates as features, which are then factored into the translation decisions. We planned to prime lexical selection by providing the results of CWSD through these features.

We planned to compare results using:

1. Hiero alone.
2. Morphological analysis on the source text.
3. WSD for lexical selection.

4.2.2. Results

4.2.2.1. WMT06

Training and test data are those used for the *NAACL 2006 Workshop Shared Task: Exploiting Parallel Texts for Statistical Machine Translation* (WMT06) (Koehn & Monz 2006). Table 15 lists the WMT06 results (test data only) for comparison, including:

- Highest: the highest reported score.
- Lowest: the lowest reported score.
- Mean, all: the mean of all scores.
- Mean, pack: the mean of all scores except the lowest two.

Columns indicate results on the following data sets:

- Dev: WMT06 development test set.
- In: WMT06 test set, in-domain.
- Out: WMT06 test set, out-of-domain.

		Fr-En			Es-En			De-En			En-Fr			En-Es			En-De		
		Dev	In	Out	Dev	In	Out	Dev	In	Out	Dev	In	Out	Dev	In	Out	Dev	In	Out
WMT06 participants	<i>Highest</i>	31.94	22.50		32.37	28.35		27.30	18.87		33.66	25.26		31.85	27.76		18.85	11.82	
	<i>Lowest</i>	21.44	19.42		23.91	19.17		15.86	11.78		25.07	21.44		23.17	16.83		9.84	6.55	
	<i>Mean, all</i>	29.33	20.70		29.99	25.77		23.75	16.47		31.04	23.45		29.36	24.47		16.62	10.43	
	<i>Mean, pack</i>	30.42	21.23		30.65	26.51		25.10	17.07		31.74	23.74		30.10	25.39		18.87	10.90	
Baseline	<i>Hiero/base</i>				25.77	25.67	24.25	21.01	20.94	15.69									
	Hiero/base (sub)				27.14	27.08	24.14												
Truncation	<i>Hiero/trunc5</i>																		
	<i>Hiero/trunc6</i>																		
Morphology	<i>Hiero/morph</i>																		
WSD	<i>Hiero/wsd</i>																		
	<i>Hiero/morph/wsd</i>																		

Table 15: WMT06 results

Results incomplete. It was perhaps interesting to note that the Hiero baseline performed somewhat below the average of WMT06 participants.

5. SUMMARY

5.1. Conclusions

Our experimental results support several conclusions.

Task I: Unsupervised morphological analysis (UMA): Our UMA algorithm dramatically reduces data complexity, with apparently negligible loss of information. UMA is applicable on a wide variety of language types, showing the most pronounced effects on morphologically complex languages.

Task II: Word sense disambiguation (WSD): Adding unannotated data significantly improves WSD results, in some cases surpassing the state-of-the-art at the time the experiments were performed. Incorporating UMA significantly improves WSD results. Combining WSD classifiers produces better results than any individual classifier.

Task III: Machine translation (MT): Results incomplete.

5.2. Future work

There are a plethora of related research directions to pursue.

5.2.1. Future work related to morphological analysis

5.2.1.1. Future work related to morphological analysis heuristics

1. Posit initial selection of morphemes.

- (a) Exclude closed class words, as identified by a POS tagger.
- 2. Morphologically analyze each word in the corpus.
 - (a) Instead of using 1-best analysis, preserve probabilistic array of all possible splits.
This would be closer to true_EM.
 - (b) Factor in Baldwin's saturation/informativeness metric for weighting candidate morphemes.
- 3. Implement framework to optimize configurable parameters.

5.2.1.2. Future work related to morphological analysis evaluation

- 1. Determine effects of naive methods (e.g. truncation) on rare events.
- 2. Investigate Baldwin's saturation/informativeness metric for evaluation.
- 3. Find or develop other stand-alone measures of complexity vs. information.
- 4. Implement Baldwin's algorithm and compare results.
- 5. Implement Freitag's algorithm and compare results.
- 6. Compare to other systems against CELEX (Baayen et al. 1995).
- 7. Devise other experiments involving CELEX, such as optimizing on Spanish and evaluating results on Italian.

5.2.2. Future work related to word sense disambiguation

5.2.2.1. Future work related to WSD with unlabeled data

1. Rerun experiments on Spanish LS data with separate dev/test sets, for fair comparison of results.
2. Determine statistical significance threshold.
3. Compare results of adding various amounts of unlabeled data to results of adding various amounts of labeled data.
4. Evaluate heuristics to predict *when* unlabeled data is likely to be most helpful.
5. Evaluate filtering heuristics to predict *which* unlabeled data is likely to be most helpful.
6. Develop methods for harvesting usable unlabeled data from monolingual corpora.

5.2.2.2. Future work related to WSD with document-level features

1. Create a data set including document-level context, perhaps from Europarl corpus.
2. Explore various document-level features, including:
 - (a) other occurrences of the target word, enforcing "one document, one sense" (Yarowsky 1995);
 - (b) other occurrences of the target word, allowing probabilistic occurrences of multiple senses in the same document;
 - (c) occurrences of semantically related words (or n-grams) not found in local context.

5.2.2.3. Future work related to WSD and morphological analysis

1. Rerun experiments on Vickrey data incorporating POS features, for a fair comparison.
2. Explore other rich source-language features, such as semantic role labeling (Yarowsky & Florian 2002; Mohammed & Pedersen 2004).
3. Perform WSD experiments using off-the-shelf commercial systems for morphological analysis of well-studied languages. Compare results to WSD with no morphological analysis, and analysis using our unsupervised system.

5.2.2.4. Other future work related to WSD

1. Evaluate system on Semeval-2007 data.
2. Evaluate system on related problems, such as diacritic restoration (Yarowsky 1994).
3. Explore other methods of classifier combination, such as voting.
4. Explore feature selection using rich source-language resources, such as POS tagging or semantic role labeling (Yarowsky & Florian 2002; Mohammed & Pedersen 2004; Vickrey et. al. 2005).
5. Explore higher n-grams and/or window sizes
6. Implement framework to optimize configurable parameters.
7. Explore clustering of source-language words using target-language senses.
(CWSD)

5.2.3. Future work related to machine translation

5.2.3.1. Future work related to MT with morphological analysis

1. Run various experiments on the language pairs identified in Task II.C, using evaluation data constructed from the Europarl corpus. All preprocessing occurs before alignment.
 - (a) Determine baseline by running MT with no morphological preprocessing.
 - (b) Determine results of truncating all source text words to 4, 5, or 6 characters.
 - (c) Determine results of running an off-the-shelf lemmatizer on the source text.
 - (d) Determine results of running an off-the-shelf morphological analyzer on the source text.
 - (e) Determine results of running our unsupervised morphological analysis system on the source text.
 - (f) (*Distant future.*) Run morphological analysis on the target text, as well. Develop a generation algorithm to reconstruct the translation from morphological components.
2. Implement Koehn & Knight's (2003) algorithm for identifying crosslingually relevant morphology.

5.2.3.2. Future work related to MT with WSD

1. Run experiments on the language pairs identified in Task II.C, using evaluation data constructed from the Europarl corpus.

- (a) Determine results of running variants of crosslingual WSD on the source text.
2. Run experiments on data from (Cabezas & Resnik 2005) and compare results.
3. Run experiments on Europarl de-en data and compare results to (Koehn & Knight 2003).

Their original data set is no longer available, per correspondence with Philipp Koehn.

4. Run experiments on GALE Arabic→English data. This would be our first set of experiments using a templatic language.
5. Make enhancements based on more recent research efforts, e.g. (Carpuat & Wu 2007a, Carpuat & Wu 2007b, Chan et al. 2007).

5.2.3.3. Future work related to MT evaluation

1. Design (or adapt) an evaluation metric to reflect adequacy of translation, based on s-score or similar measure.

6. APPENDIX: ADDITIONAL RESOURCES

6.1.1. Rule-based morphological analysis

Off-the-shelf morphological tools are available to the research community for a variety of common languages. These systems are mostly rule-based and language-specific. We would expect them to outperform anything we could induce automatically.

- English
 - Daciuk 2004
 - `file:///fs/nlp/Programs/multext_morph`
- Arabic
 - Habash & Rambow (2005)
- Bulgarian
 - HRISTO DIMITROV KRUSHKOV
- Czech
 - `file:///fs/nlp/Programs/morph_czech`
 - FMorph
- French
 - Daciuk 2004
 - ARTFL
- German
 - `file:///fs/nlp/Programs/german_morph`
 - `file:///fs/nlp/Programs/german_tools`
 - `file:///fs/nlp/Programs/multext_morph`
 - Daciuk 2004

- Hindi
 - Vasu Renganathan
 - Language Technologies Research Centre
- Italian
 - file:///fs/nlp/Programs/morph_italian
- Japanese
 - file:///fs/nlp/Programs/morphology/programs
- Kannada
 - Language Technologies Research Centre
- Marathi
 - Language Technologies Research Centre
- Polish
 - Daciuk 2004
- Punjabi
 - Language Technologies Research Centre
- Spanish
 - file:///fs/nlp/Programs/morphology/programs
- Telugu
 - Language Technologies Research Centre
- general
 - file:///fs/nlp/Programs/SIL

7. GLOSSARY

Definitions are intended for usefulness to an educated audience, not scientific precision.

1-BEST: When returning results, return only the top-scoring candidate. This is in contrast to **n-best**, where several high-scoring candidates might be returned, each with a probability (or other score).

ACCURACY: Usually used in its generic sense, equivalent to correctness; when indicated, a specific measure of a system's correctness. Contrast: precision, recall, F-score.

AFFIX: A bound morpheme, i.e. one that cannot stand alone as a word, but must be attached to a root. Affixes are often differentiated between inflectional and derivational types.

AGGLUTINATIVE: A language type (see below).

ALIGNMENT: The process of examining texts which are purported translations of each other, and determining which subparts are translations of each other. For instance, if the system is given aligned sentences, its goal is to identify aligned words. Word alignment is a subprocess of many machine translation algorithms.

ANALYTIC: A language type (see below).

ANNOTATED (OR LABELED): Data annotated with correct "answers" for whatever task is at hand. For instance, if the task is word sense disambiguation, each word in the annotated data comes labeled with its correct sense.

ATTESTED: Of data, occurring in an observed corpus. For instance, the word "reference" is attested in this document, as it appears in the following paragraph. The word "flabbergasted" is not attested in this document (or wasn't, until I wrote this sentence).

BLEU: A popular metric for evaluating the accuracy of machine translation systems, by comparing the system's output with a set of "correct" reference translations. A higher BLEU score is presumably better. Scores are not comparable between test sets, so BLEU is only useful when comparing different systems using the same test set.

CLOSED CLASS: Refers to a class of objects (typically a part-of-speech) where the entire contents of the class are known and unlikely to change. Examples: English prepositions, articles, auxiliary verbs. Contrast: open class.

COMPOUNDING: The process of combining multiple roots of the same part-of-speech into a single word. The meaning of the resulting word may have a meaning that is not determined directly from the meanings of its components. For instance, English *manhole*.

CONFIDENCE: In a statistical system, confidence is the system's own estimate of the reliability of its results.

CONTENT WORD: A word that carries significant semantic meaning, such that any translation would have to include that meaning in order to be considered accurate. These are generally open class, such as English nouns and verbs. Contrast: function word.

CORPUS (plural CORPORA): A (usually large) body of data; in our case, written text.

COTRAINING: Loosely speaking, running two independent analyses of the same data, where the output of each analysis informs the other (Yarowsky 1995; Blum & Mitchell 1998).

CROSSLINGUAL WORD SENSE DISAMBIGUATION (CWS): WSD in which a word's senses are taken to be its possible translations in another language.

CWS: See: crosslingual word sense disambiguation.

DERIVATIONAL AFFIX: An affix that is not entirely productive, may produce idiosyncratic changes in meaning, and may change the part of speech. For instance, the *-er* affix in English is derivational. The derived meanings of *farm*→*farmer* and *ministry*→*minister* are not predictable. Discarding a derivational affix almost always results in significant loss or change of meaning, unlike an inflectional affix.

DEVELOPMENT (DEV) DATA: Annotated data set aside for optimizing a system's parameters. This is still considered a "fair" system, which would not be the case if the system were optimized directly on the test data.

DIACRITIC: A symbol that appears over the top of a letter (or occasionally beneath it), such as an accent or umlaut.

EM: See: expectation-maximization.

EUROPARL CORPUS: A large parallel corpus covering 11 European languages, drawn from European Union parliamentary proceedings (Koehn 2005).

EXPECTATION-MAXIMIZATION (EM) ALGORITHM: A popular machine learning algorithm that iteratively uses current guesses about unknown values to make new guesses for the next iteration (Demster et al. 1977).

F-SCORE: A common measure of a system's correctness, calculated as the harmonic mean of precision and recall.

FORM: A word form.

FUNCTION WORD: A word that does not carry significant meaning, instead reflecting grammatical or syntactic characteristics. These are generally closed class, such as English prepositions. Contrast: content word.

FUSIONAL: A language type (see below).

HAPAX LEGOMENON: A word form that occurs precisely once in a corpus.

INFLECTIONAL AFFIX: An affix that is highly productive and regular in its use. They generally add only grammatical information, and rarely change the part of speech. For instance, the *-s* affix in English is inflectional. An inflectional affix can often be discarded without significant loss of meaning, unlike a derivational affix.

IRREGULAR: See: regular.

LABELED: See: annotated.

LEMMA: Another name for a root.

LEMMATIZATION: The process of replacing each word with its lemma(s), typically by identifying and discarding affixes. For example, lemmatizing the sentence, "the biggest dogs ate the smallest meals," yields, "the big dog eat the small meal".

MACHINE TRANSLATION (MT): The fully automated process of translating text from one language to another.

MORPHEME: Informally, the smallest discernible unit of meaning. A word is typically composed of one or more morphemes, differentiated between roots and affixes. For instance, the English word *farmers* can be decomposed into three morphemes: *farm* *-er* *-s*, where *farm* is the root, and *-er* and *-s* are affixes.

There is evidence that morphemes are composed of yet smaller units of meaning. However, just as many scientists are happy to act as though atoms are the smallest units of matter, so we will maintain our happy delusion that morphemes are atomic.

MORPHOLOGICAL ANALYSIS: The process of breaking a word, or all the words of a given language, into component morphemes.

MORPHOLOGICALLY POOR: Having comparatively few inflectional affixes and/or irregular forms. English is morphologically poor, having comparatively few word endings.

MORPHOLOGICALLY RICH: Having comparatively many inflectional affixes and/or irregular forms. Czech is morphologically rich, having different word endings for gender, person, number, case, and phase of the moon, in all different combinations. (Well, I'm not sure about the last one.)

MT: See: machine translation.

N-GRAM: A string of n characters.

NOISE: In the context of a statistical application, *noise* refers to annotated data where the annotations are less than 100% accurate – a virtually universal condition of real-world data. This can lead a naive system to draw incorrect conclusions. A robust system either filters out the noise, or otherwise avoids weighting it too heavily.

OPEN CLASS: Refers to a class of objects (typically a part-of-speech) where the contents of the class cannot be readily enumerated, and change frequently. Examples: English nouns, verbs, adjectives.

PARALLEL CORPUS (OF PARALLEL TEXT): A corpus of text that has been translated directly between two (or more) languages. Typically, such a corpus is initially aligned only at the document level. i.e. It is not marked which sentences or words are translations of each other.

PBSMT: See: phrase-based statistical machine translation.

PHRASE-BASED STATISTICAL MACHINE TRANSLATION (PBSMT): A form of machine translation which translates a sequence of words ("phrase") in the source language to a sequence of words in the target language as a single operation. Contrast: WBSMT.

POS: Part-of speech: noun, verb, adjective, etc.

PRECISION: A common measure of a system's correctness, calculated as: $(\text{number of correct identifications}) / (\text{number of identifications produced by the system})$. So, if there were 100 possible correct answers, but the system only offered 10 guesses, of which 6 were correct, the precision is: $(6 / 10) = 60\%$. Contrast: recall, F-score.

PRODUCTIVE: Refers to a process that can be applied in most situations, even novel cases. For instance, in English, applying *-s* to form a plural noun or *-ing* to form a progressive verb are highly productive processes. Applying *-tion* to form a noun from a verb is not productive, applying only in a relatively small number of cases.

RECALL: A common measure of a system's correctness, calculated as: $(\text{number of correct identifications}) / (\text{number of potential correct identifications})$. So, if there were 100 possible correct answers, but the system only offered 10 guesses, of which 6 were correct, the recall is: $(6 / 100) = 6\%$. Contrast: precision, F-score.

REGULAR: Refers to a process that follows a typical productive pattern. For instance, in English, forming *cat + -s -> cats* is regular pluralization. Forming *man + -s -> men* is irregular pluralization. The most common words are the most likely to follow irregular patterns, while novel words almost exclusively follow regular patterns (Pinker 1999).

ROBUST : See: noise.

ROOT (or LEMMA): An unbound morpheme, i.e. one that can stand alone as a word. Contrast affix.

RULE-BASED: A system built around rules written by experts, rather than statistical machine learning.

SEMI-SUPERVISED: A statistical system that relies for training on a small amount of annotated data and a large amount of unannotated data. Contrast: supervised, unsupervised.

SENSE: A possible meaning for a given word. For example, the English word *fence* has at least three senses:

1. He climbed over the *fence*.
2. He learned to *fence* with a sabre.
3. He went into the city to *fence* the stolen TV's.

SENSEVAL: Senseval (now Semeval) is a series of workshops designed to test and showcase WSD systems. The first workshop, Senseval-1, was held in 1998. The most recent, Semeval-1/Senseval-4, was held in 2006.

SOURCE TEXT: In a translation exercise, the input text that needs to be translated. The source language is the language of this text. Contrast: target text.

SPARSE: Data that does not well represent all possibilities. Ideally, you want your data to contain an example of every possible event you might see. However, this is only possible in a finite domain. If you're looking, for instance, at all possible uses of the word *fence*, you want at least several hundred sentences, if not several thousand, to capture the variety of possible contexts. The fewer such sentences you have, the greater your data sparseness. A major challenge of designing a statistical system is dealing with data sparseness.

STEM: In this discussion, a stem is any word without inflectional affixes, containing one or more roots and zero or more derivational affixes. Specifically, a word's stem refers to the largest possible stem in that word. So, for the English word *farmers*, its root is *farm*, and its stem is *farmer*.

STEMMING: The process of replacing each word with its stem (or root), typically by identifying and discarding inflectional (or all) affixes.

SUPERVISED: A statistical system that relies for training on large amounts of annotated data, often requiring considerable time and effort by experts to annotate the data. Contrast: unsupervised, semi-supervised.

SUPPLETIVE: A language type (see below).

TARGET TEXT: The output of translation. The target language is the language of this text. Contrast: source text.

TEMPLATIC: A language type (see below).

TEST DATA: Annotated data set aside for testing a system's performance, For fair results, the test set must not be observed while building the system. Contrast: dev data.

TOKEN: In this discussion, a token is any occurrence of a string of non-punctuation, non-space characters. The precise definition varies between applications.

TRAINING DATA: Annotated data used to train a statistical system.

UMA: See: unsupervised morphological analysis.

UNANNOTATED: Raw text that has not been annotated for any particular purpose, such as might come directly from a news broadcast or internet post.

UNSUPERVISED: A statistical system that relies for training only on unannotated data, often readily available in large amounts. Contrast: supervised, semi-supervised.

UNSUPERVISED MORPHOLOGICAL ANALYSIS (UMA): A system that performs morphological analysis without requiring experts to write rules or break words into morphemes by hand.

WBSMT: See: word-based statistical machine translation.

WORD: In this discussion, the term *word* typically refers to a word form, although it may sometimes be used to indicate a token. The intended usage should be obvious from context.

WORD FORM: A class of tokens with identical spelling. For instance, the sentence "the big cat chased the big dog" contains seven tokens, but only five word forms.

WORD SENSE DISAMBIGUATION (WSD): The process of determining which possible sense of a word is used in a given context.

WORD-BASED STATISTICAL MACHINE TRANSLATION (WBSMT): A form of machine translation which translates each word in the source language to a word in the target language as a single operation. Contrast: PBSMT.

ZIPF'S LAW: George Kingsley Zipf observed a property of language that the frequency of any word in a corpus is inversely proportional to its rank in the frequency table. Thus, the most frequent word occurs approximately twice as often as the second most frequent word, which occurs approximately twice as often as the fourth most frequent word, and so on. At the other end of the curve, the "long tail", rare events are by far the highest class of word forms, with hapax legomena being the highest class of all. (*I haven't been able to confirm whether the same ratio applies.*) Zipf's law has since been observed in a large number of natural (and unnatural) distributions.

7.1.1.1. Language types

Languages are often grouped into three main classes.

- An ANALYTIC (OR ISOLATING) language typically only has one morpheme per word.
- An AGGLUTINATIVE (OR POLYSYNTHETIC) language typically has words made up of many distinct roots and affixes, where each root or affix carries a single unit of meaning.
Example: German [*Dämpfungsscheibenanordnung*] 'dampening disk assembly'
Technically, a polysynthetic language often has a single word representing a whole sentence, while an agglutinative language does not. However, we do not make use of this distinction.
- A FUSIONAL language typically has words made up of a root and one or more affixes, where a single affix can carry several units of meaning.
Example: French [*étudiant, étudiante, étudiants, étudiantes*] 'student'.

We use these additional subclassifications in our discussion.

- A SUPPLETIVE language is a fusional language with a high proportion of words which take on a whole new form when inflected, as in English [*go + PAST*] → *went*.
- A TEMPLATIC language is a fusional language where the root (or stem) is a sequence of characters (typically consonants) and the affix is another sequence of characters/diacritics (typically vowels) which is interweaved into the root.

In reality, a language rarely fits precisely into one class. For instance, English is predominately analytic ("have been going"), but has some fusional ("he dance-s"), suppletive ("he went"), and even agglutinative ("garbage-truck-windshield-wiper-cleaner"), characteristics.

8. REFERENCES

- Galen Andrew, Trond Grenager, and Christopher Manning. 2004. **Verb Sense and Subcategorization: Using Joint Inference to Improve Performance on Complementary Tasks**. *EMNLP 2004*: 150-157.
- R.H. Baayen, R. Piepenbrock, and L. Gulikers. 1995. *The CELEX Lexical Database* (CD-ROM). LDC, University of Pennsylvania, Philadelphia..
- Bogdan Babych and Anthony Hartley. 2003. **Selecting Translation Strategies in MT using Automatic Named Entity Recognition**. *European Association for Machine Translation*.
- Timothy Baldwin. 2005. **Bootstrapping Deep Lexical Resources: Resources for Courses**. *Proceedings of the ACL-SIGLEX 2005 Workshop on Deep Lexical Acquisition*: 67–76. Ann Arbor, USA.
- J. Bilmes and K. Kirchhoff. 2003. Factored Language Models and Generalized Parallel Backoff. *Human Language Technology Conference*.
- A. Blum and T. Mitchell. 1998. **Combining Labeled and Unlabeled Data with Co-Training**. *Proceedings of the 1998 Conference on Computational Learning Theory*. July 1998.
- Clara Cabezas and Philip Resnik. 2005. **Using WSD Techniques for Lexical Selection in Statistical Machine Translation**. *Technical Report: LAMP-TR-124/CS-TR-4736/UMIACS-TR-2005-42*. University of Maryland, College Park, July 2005.

- Marine Carpuat and Dekai Wu. 2007a. **How Phrase Sense Disambiguation outperforms Word Sense Disambiguation for Statistical Machine Translation.** *11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2007)*. Skovde: Sep 2007.
- Marine Carpuat and Dekai Wu. 2007b. **Improving Statistical Machine Translation using Word Sense Disambiguation.** *2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*. Prague: Jun 2007.
- Marine Carpuat and Dekai Wu. 2005. **Word Sense Disambiguation vs. Statistical Machine Translation.** *43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005)*. Ann Arbor, MI, Jun 2005.
- Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. **Word sense disambiguation improves statistical machine translation.** *Proc. ACL*.
- David Chiang. 2005. **A Hierarchical Phrase-Based Model for Statistical Machine Translation.** *Proceedings of ACL 2005*: 263–270. Best paper award.
- I. Chugur, J. Gonzalo, F. Verdejo. 2000. **Sense distinctions in NLP applications.** *Proceedings of Ontolex*.
- Martin Čmejrek, Jan Cuřín, Jiří Havelka, Jan Hajič, Vladislav Kuboň. 2004. **Prague Czech-English Dependency Treebank: Syntactically Annotated Resources for Machine Translation.** *4th International Conference on Language Resources and Evaluation*. Lisbon, Portugal.
- Jan Daciuk. 2004. **Finite-State Lexical Tools.** *BIS 2004, 7th International Conference on Business Information Systems*: 373-380. Witold Abramowicz (ed.), Wydawnictwo Akademii Ekonomicznej w Poznaniu, Poznań, Poland, 21-23 April, 2004.

- Jan Daciuk and Gertjan van Noord. 2001. **A Finite-State Library for NLP**. *CLIN 2001*. University of Twente, Enschede, the Netherlands, November 2001.
- A. de Gispert. 2005. **Phrase Linguistic Classification and Generalization for Improving Statistical Machine Translation**. *Proc. of the ACL Student Research Workshop (ACL'05/SRW)*: 67-72. Ann Arbor (Michigan), June 2005.
- Arthur Dempster, Nan Laird, and Donald Rubin. 1977. **Maximum likelihood from incomplete data via the EM algorithm**. *Journal of the Royal Statistical Society, Series B*, 39(1): 1–38.
- R. Florian and R. Wicentowski. 2002. **Unsupervised italian word sense disambiguation using wordnets and unlabeled corpora**. *Proceedings of SigLEX'02*: 67–73.
- Dayne Freitag. **Morphology Induction from Term Clusters**. *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*: 128-135. Ann Arbor, MI, June 2005.
- Sharon Goldwater, David McClosky. 2005. **Improving Statistical MT through Morphological Analysis**. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Vancouver.
- Nizar Habash, Owen Rambow. 2005. **Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop**. *ACL-05*.
- D. Hiemstra. 1996. **Using Statistical Methods to Create a Bilingual Dictionary**. *Master's thesis*. Universiteit Twente.

- Anas E. Isbihani, Shahram Khadivi, Oliver Bender, and Hermann Ney. 2006. **Morpho-syntactic Arabic Preprocessing for Arabic to English Statistical Machine Translation**. *Proceedings on the Workshop on Statistical Machine Translation*: 15-22. New York City, June 2006.
- T. Joachims. 2003. **Transductive Learning via Spectral Graph Partitioning**. *International Conference on Machine Learning (ICML)*.
- T. Joachims. 1999. **Making large-Scale SVM Learning Practical**. *Advances in Kernel Methods - Support Vector Learning*. B. Schölkopf and C. Burges and A. Smola (ed.). MIT-Press.
- Martin Kay. 1987. **Nonconcatenative Finite-State Morphology**. *EACL 1987*: 2-10.
- Edward Kenschaf. 2005. **Improving Crosslingual Word Sense Disambiguation using Unlabeled Monolingual Corpora**. Unpublished.
- K. Kirchhoff and M. Yang. 2005. **Improved Language Modeling for Statistical Machine Translation**, *Proceedings of the ACL Workshop on Building and Using Parallel Texts*.
- Philipp Koehn. 2005. **Europarl: A Parallel Corpus for Statistical Machine Translation**. *MT Summit 2005*.
- Philipp Koehn. 2004. **Pharaoh: A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models**. *AMTA 2004*.
- Philipp Koehn. 2002. **Europarl: A Multilingual Corpus for Evaluation of Machine Translation**. Draft, unpublished.

- Philipp Koehn and Kevin Knight. 2003. **Empirical Methods for Compound Splitting.** *EACL 2003.*
- Philipp Koehn and Christof Monz. 2006. **Manual and Automatic Evaluation of Machine Translation between European Languages.** *Proceedings on the Workshop on Statistical Machine Translation.* June 2006.
- Philipp Koehn, Franz Josef Och, Daniel Marcu. 2003. **Statistical Phrase-Based Translation.** *Proceedings of the Human Language Technology Conference 2003 (HLT-NAACL 2003).* Edmonton, Canada, May 2003.
- Kimmo Koskenniemi. 1983. **Two-level Morphology: A General Computational Model for Word-Form Recognition and Production.** *Publications*, p. 160. University of Helsinki, Department of General Linguistics.
- Young-Suk Lee. **Morphological Analysis for Statistical Machine Translation.** *HLT-NAACL 2004: Short Papers: 57–60.* Susan Dumais, Daniel Marcu, Salim Roukos, eds. Boston, Massachusetts, May 2004.
- Young-Suk Lee, Kishore Papineni, Salim Roukos, Ossama Emam, Hany Hassan. 2003. **Language Model Based Arabic Word Segmentation.** *Proceedings of the 41st Annual Meeting of the ACL: 399-406.* Sapporo, Japan.
- L. Màrquez, M. Taulé, M.A. Martí, M. García, N. Artigas, F. Real, D. Ferrés. 2004. **Senseval-3: The Spanish Lexical Sample Task.** *Proceedings of the Senseval-3 ACL-SIGLEX Workshop.* Barcelona, Spain.
- Martin, Mihalcea and Pedersen. 2005. **Word Alignment for Languages with Scarce Resources.** *Proceedings of the ACL Workshop on Building and Using Parallel Texts.* Ann Arbor, MI, June 29-30.

- Rada Mihalcea and Phil Edmonds, ed. 2004. *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Association for Computational Linguistics: Barcelona, Spain, July, 2004.
- Mohammad and Pedersen. 2004. **Combining Lexical and Syntactic Features for Supervised Word Sense Disambiguation**. *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*. Boston, MA, May 6-7, 2004.
- Sonja Nießen, Hermann Ney. 2004. **Statistical Machine Translation with Scarce Resources Using Morpho-Syntactic Information**. *Computational Linguistics*, Volume 30, Number 2: 181-204, June 2004.
- Sonja Nießen, Hermann Ney. 2001a. **Toward hierarchical models for statistical machine translation of inflected languages**. *ACL-EACL-2001: 39th Annual Meeting of the Association for Computational Linguistics - joint with EACL 2001: Proceedings of the Workshop on Data-Driven Machine Translation*: 47-54. Toulouse, France, July 2001.
- Sonja Nießen, Hermann Ney. 2001b. **Morpho-syntactic analysis for Reordering in Statistical Machine Translation**. *Proceedings of the MT Summit VIII*: 247-252. Santiago de Compostela, Galicia, Spain, September 2001.
- Sonja Nießen, Hermann Ney. 2000. **Improving SMT Quality with Morpho-Syntactic Analysis**. *Proceedings of the 20th International Conference on Computational Linguistics*: 1081-1085. Saarbrücken, Germany.
- Franz Josef Och. 1999. **An Efficient Method for Determining Bilingual Word Classes**. *Ninth Conf. of the Europ. Chapter of the Association for Computational Linguistics (EACL'99)*: 71-76. Bergen, Norway, June 1999.

- Franz Josef Och and Hermann Ney. 2004. **The Alignment Template Approach to Statistical Machine Translation**. *Computational Linguistics* 30: 417–449.
- Franz Josef Och and Hermann Ney. 2003. **A Systematic Comparison of Various Statistical Alignment Models**. *Computational Linguistics* 29(1): 19-51. March 2003.
- Thanh Phong Pham, Hwee Tou Ng, and Wee Sun Lee. 2005. **Word Sense Disambiguation with Semi-Supervised Learning**. *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*: 1093-1098. Pittsburgh, Pennsylvania, USA.
- Steven Pinker. 1999. *Words and Rules: The Ingredients of Language*. HarperCollins.
- Maja Popović and Hermann Ney. 2004a. **Improving Word Alignment Quality using Morpho-syntactic Information**. *COLING04*.
- Maja Popović and Hermann Ney. 2004b. **Towards the Use of Word Stems and Suffixes for Statistical Machine Translation**. *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*: 1585-1588. Lisbon, Portugal, May 2004.
- Maja Popović, David Vilar, Hermann Ney, Slobodan Jovičić, Zoran Šarić. 2005. **Augmenting a Small Parallel Text with Morpho-syntactic Language Resources for Serbian-English Statistical Machine Translation**. *Proceedings of the ACL Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*: 41-48. Ann Arbor, Michigan, June 2005.
- Philip Resnik and David Yarowsky. 2000. **Distinguishing Systems and Distinguishing Senses: New Evaluation Methods for Word Sense Disambiguation**. *Natural Language Engineering*, 5(2): 113-133.

- Bettina Schrader. 2006. **Non-Probabilistic Alignment of Rare German and English Nominal Expressions**. *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*. Genoa, Italy, May 2006.
- Bettina Schrader. 2004. **Improving Word Alignment Quality Using Linguistic Knowledge**. *Proceedings of the Workshop on The Amazing Utility of Parallel and Comparable Corpora (LREC 2004 satellite event)*: 46-49. Lisbon, Portugal, May 2004.
- David Vickrey, Luke Biewald, Marc Teysier, and Daphne Koller. 2005. **Word-Sense Disambiguation for Machine Translation**. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*: 771-778. Vancouver, British Columbia, Canada, October 2005.
- Richard Wicentowski. 2002. **Modeling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework**. Doctoral dissertation. Johns Hopkins University, Baltimore, Maryland, October 2002.
- David Yarowsky. 2000. **Hierarchical Decision Lists for Word Sense Disambiguation**. *Computers and the Humanities* 34(2): 179-186.
- David Yarowsky. 1995. **Unsupervised Word Sense Disambiguation Rivaling Supervised Methods**. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*: 189-196. Cambridge, MA.
- David Yarowsky. 1994. **Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French**. *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*: 88-95. Las Cruces, NM.
- David Yarowsky and R. Florian. 2002. **Evaluating Sense Disambiguation Performance Across Diverse Parameter Spaces**. *Journal of Natural Language Engineering*.

David Yarowsky and Richard Wicentowski. 2001. **Minimally supervised morphological analysis by multimodal alignment**. *Proceedings of ACL-01*.

David Yarowsky and Richard Wicentowski. 2000. **Minimally supervised morphological analysis by multimodal alignment**. *Proceedings of ACL-2000*: 207-216. Hong Kong.