Web-Scale N-gram Models for Lexical Disambiguation.

Conference Paper · January 2009

Source: DBLP

3 authors, including:

Shane Bergsma
University of Saskatchewan
33 PUBLICATIONS **433** CITATIONS
SEE PROFILE

Randy Goebel
University of Alberta
150 PUBLICATIONS **1,214** CITATIONS
SEE PROFILE

Web-Scale N-gram Models for Lexical Disambiguation

Shane Bergsma
Department of Computing Science
University of Alberta
Edmonton, Alberta
Canada, T6G 2E8
bergsma@cs.ualberta.ca

Dekang Lin
Google, Inc.
1600 Amphitheatre Parkway
Mountain View
California, 94301
lindek@google.com

Randy Goebel
Department of Computing Science
University of Alberta
Edmonton, Alberta
Canada, T6G 2E8
goebel@cs.ualberta.ca

Abstract

Web-scale data has been used in a diverse range of language research. Most of this research has used web counts for only short, fixed spans of context. We present a unified view of using web counts for lexical disambiguation. Unlike previous approaches, our supervised and unsupervised systems combine information from multiple and overlapping segments of context. On the tasks of preposition selection and context-sensitive spelling correction, the supervised system reduces disambiguation error by 20-24% over the current state-of-the-art.

1 Introduction

Many problems in Natural Language Processing (NLP) can be viewed as assigning labels to particular words in text, given the word's context. If the decision process requires choosing a label from a predefined set of possible choices, called a *candidate set* or *confusion set*, the process is often referred to as *disambiguation* [Roth, 1998]. Part-of-speech tagging, spelling correction, and word sense disambiguation are all lexical disambiguation processes.

One common disambiguation task is the identification of word-choice errors in text. A language checker can flag an error if a confusable alternative better fits a given context:

(1) The system tried to decide {*among, between*} the two confusable words.

Most NLP systems resolve such ambiguity with the help of a large corpus of text. The corpus indicates which candidate is more frequent in similar contexts. The larger the corpus, the more accurate the disambiguation [Banko and Brill, 2001]. Since no corpus is as large as the world wide web, many systems incorporate web counts into their selection process. For the above example, a typical web-based system would query a search engine with the sequences "decide *among* the" and "decide *between* the" and select the candidate that returns the most pages [Lapata and Keller, 2005]. Clearly, this approach fails when more context is needed for disambiguation.

We present a unified view of web-scale approaches to lexical disambiguation. Rather than using a single context sequence, we use contexts of various lengths and positions. There are five 5-grams, four 4-grams, three trigrams and two bigrams spanning the target word in Example (1). We gather counts for each of these sequences, with each candidate in the target position. We first show how the counts can be used as features in a supervised classifier, with a count's contribution weighted by its context's size and position. We also propose a novel unsupervised system that simply sums a subset of the (log) counts for each candidate. Surprisingly, this system achieves most of the gains of the supervised approach without requiring any training data. Our systems outperform traditional web-scale approaches on the tasks of preposition selection, context-sensitive spelling correction, and non-referential pronoun detection.

2 Related Work

Yarowsky [1994] defines lexical disambiguation as a task where a system must "disambiguate two or more semantically distinct word-forms which have been conflated into the same representation in some medium." Lapata and Keller [2005] divide disambiguation problems into two groups: generation and analysis. In generation, the confusable candidates are actual words, like *among* and *between*. In analysis, we disambiguate semantic labels, such as part-of-speech tags, representing abstract properties of surface words.

For generation tasks, a model of each candidate's distribution in text is created. The models indicate which usage best fits each context, enabling candidate disambiguation in tasks such as spelling correction [Golding and Roth, 1999], preposition selection [Chodorow *et al.*, 2007; Felice and Pulman, 2007], and diacritic restoration [Yarowsky, 1994]. The models can be large-scale classifiers or standard N-gram language models (LMs). Trigram LMs have long been used for spelling correction, an approach sometimes referred to as the Mays, Damerau, and Mercer model [Wilcox-O'Hearn *et al.*, 2008]. Gamon et al. [2008] use a Gigaword 5-gram LM for preposition selection. While web-scale LMs have proved useful for machine translation [Brants *et al.*, 2007], most web-scale disambiguation approaches compare specific sequence counts rather than full-sentence probabilities.

In analysis problems such as part-of-speech tagging, it is not as obvious how a LM can be used to score the candidates, since LMs do not contain the candidates themselves, only surface words. However, large LMs can also benefit these applications, provided there are surface words that correlate with the semantic labels. Essentially, we devise some surrogates

for each label, and determine the likelihood of these surrogates occurring with the given context. For example, Mihalcea and Moldovan [1999] perform sense disambiguation by creating label surrogates from similar-word lists for each sense. To choose the sense of *bass* in the phrase "caught a huge bass," we might consider *tenor*, *alto*, and *pitch* for sense one and *snapper*, *mackerel*, and *tuna* for sense two. The sense whose group has the higher web-frequency count in *bass*'s context is chosen. Similarly, Bergsma et al. [2008] identify whether the English pronoun *it* refers to a preceding noun ("*it* was hungry") or is used as a grammatical placeholder ("*it* is important to...") by testing the frequency of other words in place of *it* in the context. Since "*he* was hungry" is attested in the corpus but "*he* is important to" is not, we conclude the first instance is referential but the second is not.

Bergsma et al. [2008] also use learning to weight the counts of different context sizes and positions. Their technique was motivated and evaluated only for (binary) non-referential pronoun detection; we present a multi-class classification algorithm for general lexical disambiguation problems, and evaluate it on both generation and analysis tasks. We also show that a simple unsupervised system is competitive with supervised approaches requiring thousands of training examples.

3 Disambiguation with N-gram Counts

For a word in text, $\mathbf{w_0}$, we wish to assign a label, y_i, from a fixed set of candidates, $Y = \{y_1, y_2 ..., y_{|Y|}\}$. Assume that our target word $\mathbf{w_0}$ occurs in a sequence of context tokens: $\mathbf{W}$=$\{w_{-4}, w_{-3}, w_{-2}, w_{-1}, \mathbf{w_0}, w_1, w_2, w_3, w_4\}$. The key to improved web-scale models is that they make use of a variety of context segments, of different sizes and positions, that span the target word $\mathbf{w_0}$. We follow Bergsma et al. [2008] in calling these segments *context patterns*. The words that replace the target word are called *pattern fillers*. Let the set of pattern fillers be denoted by $F = \{f_1, f_2, ..., f_{|F|}\}$. Recall that for generation tasks, the filler set will usually be identical to the set of labels (e.g., for word selection tasks, F=Y=$\{$*among,between*$\}$). For analysis tasks, we must use other fillers, chosen as surrogates for one of the semantic labels (e.g. for WSD of *bass*, Y=$\{Sense1, Sense2\}$, F=$\{$*tenor,alto,pitch,snapper,mackerel,tuna*$\}$).

Each length-N context pattern, with a filler in place of $\mathbf{w_0}$, is an N-gram, for which we can retrieve a count. We retrieve counts from the web-scale Google Web 5-gram Corpus, which includes N-grams of length one to five.[1] For each target word $\mathbf{w_0}$, there are five 5-gram context patterns that may span it. For Example (1) in Section 1, we can extract the following 5-gram patterns:

system tried to decide $\mathbf{w_0}$
tried to decide $\mathbf{w_0}$ the
to decide $\mathbf{w_0}$ the two
decide $\mathbf{w_0}$ the two confusable
$\mathbf{w_0}$ the two confusable words

Similarly, there are four 4-gram patterns, three 3-gram patterns and two 2-gram patterns spanning the target. With $|F|$ fillers, there are $14|F|$ filled patterns with relevant N-gram counts. Here, F=$\{among, between\}$, so 28 counts are used.

[1] Available from the LDC as LDC2006T13.

3.1 SUPERLM

We use supervised learning to train a classifier, h, to map a target word and its context to a label, $h : \mathbf{W} \rightarrow Y$. Examples are represented by features, $\mathbf{\Phi}(\mathbf{W})$. The learning algorithm uses training examples to choose a set of weights, $\boldsymbol{\lambda}^y$, for each label, such that the weighted sum of the true label's features is higher than for other candidates. At test time, the highest-scoring label is chosen:

$$h(\mathbf{W}) = \underset{y \in Y}{\operatorname{argmax}} \boldsymbol{\lambda}^y \cdot \mathbf{\Phi}(\mathbf{W}) \quad (1)$$

We use features for the logarithm of each of the $14|F|$ different counts. The weight on a count depends on the class (label), the filler, the context position and its size, for a total of $14|F||Y|$ count-weight parameters. For generation tasks, the classifier tends to learn positive weight on features where y=f, with higher absolute weights on the most predictive positions and lengths. If a pattern spans outside the current sentence (when $\mathbf{w_0}$ is close to the start or end), we use zero for the corresponding feature value, but fire an indicator feature to flag that the pattern crosses a boundary.[2] We call this approach SUPERLM because it is SUPERvised, and because, like an interpolated language model (LM), it mixes N-gram statistics of different orders to produce an overall score for each filled context sequence.

SUPERLM's features differ from previous lexical disambiguation feature sets. In previous systems, attribute-value features flag the presence or absence of a particular word, part-of-speech, or N-gram in the vicinity of the target [Roth, 1998]. Hundreds of thousands of features are used, and pruning and scaling are key issues [Carlson *et al.*, 2001]. Performance scales logarithmically with the number of examples, even up to one billion training examples [Banko and Brill, 2001]. In contrast, SUPERLM's features are all aggregate counts of events in an external (web) corpus, not specific attributes of the current example. It has only $14|F||Y|$ parameters, for the weights assigned to the different counts. Much less training data is needed to achieve peak performance.

3.2 SUMLM

We create an unsupervised version of SUPERLM. We produce a score for each *filler* by summing the (unweighted) log-counts of all context patterns using that filler. For generation tasks, the filler with the highest score is taken as the label. We refer to this approach in our experiments as SUMLM. It can be shown that SUMLM is similar to a Naive Bayes classifier, but without counts for the class prior.

3.3 TRIGRAM

Previous web-scale approaches are also unsupervised. Most use one context pattern for each filler: the trigram with the filler in the middle: $\{w_{-1}, f, w_1\}$. $|F|$ counts are needed for each example, and the filler with the most counts is taken as

[2] Other features are possible. For generation tasks, we could also include synonyms of the labels as fillers. Features could also be created for counts of patterns processed in some way (e.g. converting one or more context tokens to wildcards, POS-tags, lower-case, etc.), provided the same processing can be done to the N-gram corpus.

the label [Lapata and Keller, 2005; Liu and Curran, 2006; Felice and Pulman, 2007]. Using only one count for each label is usually all that is feasible when the counts are gathered using an Internet search engine, which limits the number of queries that can be retrieved. With limited context, and somewhat arbitrary search engine page counts, performance is limited. Web-based systems are regarded as "baselines" compared to standard approaches [Lapata and Keller, 2005], or, worse, as scientifically unsound [Kilgarriff, 2007]. Rather than using search engines, higher accuracy and reliability can be obtained using a large corpus of automatically downloaded web documents [Liu and Curran, 2006]. We evaluate the trigram pattern approach, with counts from the Google 5-gram corpus, and refer to it as TRIGRAM in our experiments.

3.4 RATIOLM

Carlson et al. [2008] proposed an unsupervised method for spelling correction that also uses counts for various pattern fillers from the Google 5-gram Corpus. For every context pattern spanning the target word, the algorithm calculates the ratio between the highest and second-highest filler counts. The position with the highest ratio is taken as the "most discriminating," and the filler with the higher count in this position is chosen as the label. The algorithm starts with 5-grams and backs off to lower orders if no 5-gram counts are available. This position-weighting (*viz.* feature-weighting) technique is similar to the decision-list weighting in [Yarowsky, 1994]. We refer to this approach as RATIOLM in our experiments.

4 Applications

While all disambiguation problems can be tackled in a common framework, most approaches are developed for a specific task. Like Roth [1998], we take a unified view of disambiguation, and apply our systems to preposition selection, spelling correction, and non-referential pronoun detection.

4.1 Preposition Selection

Choosing the correct preposition is one of the most difficult tasks for a second-language learner to master, and errors involving prepositions constitute a significant proportion of errors made by learners of English [Chodorow *et al.*, 2007].

Several automatic approaches to preposition selection have recently been developed [Felice and Pulman, 2007; Gamon *et al.*, 2008]. We follow the experiments of Chodorow et al. [2007], who train a classifier to choose the correct preposition among 34 candidates.[3] In [Chodorow *et al.*, 2007], feature vectors indicate words and part-of-speech tags near the preposition, similar to the features used in most disambiguation systems, and unlike the aggregate counts we use in our supervised preposition-selection N-gram model (Section 3.1).

For preposition selection, like all generation disambiguation tasks, labeled data is essentially free to create. Each preposition in edited text is assumed to be correct, automatically providing an example of that preposition's class. We extract examples from the New York Times (NYT) section of the Gigaword corpus.[4] We take the first 1 million prepositions in NYT as a training set, 10K from the middle as a development set and 10K from the end as a final unseen test set. We tokenize the corpus and identify prepositions by string-match. Our system uses no parsing or part-of-speech tagging to extract the examples or create the features.

4.2 Context-sensitive Spelling Correction

We also evaluate on the classic generation problem of context-sensitive spelling correction. For every occurrence of a word in a pre-defined confusion set (like {*among, between*}), we select the most likely word from the set. The importance of using large volumes of data has previously been noted [Banko and Brill, 2001; Liu and Curran, 2006]. Impressive levels of accuracy have been achieved on the standard confusion sets, for example, 100% on disambiguating both {*affect, effect*} and {*weather, whether*} by Golding and Roth [1999]. We thus restricted our experiments to the five confusion sets (of twenty-one in total) where the reported performance in [Golding and Roth, 1999] is below 90% (an average of 87%): {*among, between*}, {*amount, number*}, {*cite, sight, site*}, {*peace, piece*}, and {*raise, rise*}. We again create labeled data automatically from the NYT portion of Gigaword. For each confusion set, we extract 100K examples for training, 10K for development, and 10K for a final test set.

4.3 Non-referential Pronoun Detection

We can cast Bergsma et al. [2008]'s approach to non-referential pronoun detection as an instance of SUPERLM. They use fillers: F = {the pronoun *it*, the pronoun *they*, other pronouns, the ⟨UNK⟩ token, and all other tokens (*all*)}. The classifier learns the relation between the filler counts and the two labels (Y={$Ref, NonRef$}). Relatively higher counts for the *it*-filler generally indicate a non-referential instance.

We extend their work by applying our full set of web-scale models. For SUMLM, we decide $NonRef$ if the difference between the SUMLM scores for *it* and *they* is above a threshold. For TRIGRAM, we threshold the ratio between *it*-counts and *they*-counts. For RATIOLM, we compare the frequencies of *it* and *all*, and decide $NonRef$ if the count of *it* is higher. The thresholds and comparisons are optimized on the dev set.

We preprocessed the N-gram corpus exactly as described in [Bergsma *et al.*, 2008], and used the same portion of *It-Bank* evaluation data.[5] We take the first half of each of the subsets for training, the next quarter for development and the final quarter for testing, creating an aggregate set with 1070 training, 533 development and 534 test examples.

4.4 Evaluation Methodology

We evaluate using *accuracy*: the percentage of correctly-selected labels. As a baseline (BASE), we state the accuracy of always choosing the most-frequent class. For spelling correction, we average accuracies across the five confusion sets.

[3] Chodorow et al. do not identify the 34 prepositions they use. We use the 34 from the SemEval-07 preposition sense-disambiguation task [Litkowski and Hargraves, 2007]: *about, across, above, after, against, along, among, around, as, at, before, behind, beneath, beside, between, by, down, during, for, from, in, inside, into, like, of, off, on, onto, over, round, through, to, towards, with*

[4] Available from the LDC as LDC2003T05

[5] Available at www.cs.ualberta.ca/~bergsma/ItBank/

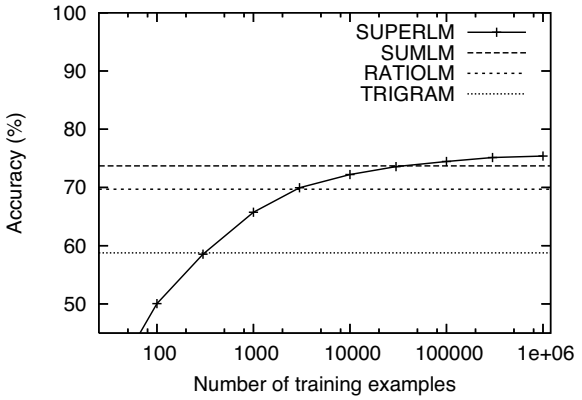

Figure 1: Preposition selection learning curve

We also provide learning curves by varying the number of labeled training examples. It is worth reiterating that this data is used solely to weight the contribution of the different filler counts; the filler counts themselves do not change, as they are always extracted from the full Google 5-gram Corpus.

SUPERLM uses a linear-kernel multiclass SVM (the efficient SVM^{multiclass} instance of SVM^{struct} [Tsochantaridis *et al.*, 2004]). It slightly outperformed one-versus-all SVMs in preliminary experiments. We tune the SVM's regularization on the development sets. We apply add-one smoothing to the counts used in SUMLM and SUPERLM, while we add 39 to the counts in RATIOLM, following the approach of Carlson et al. [2008] (40 is the count cut-off used in the Google Corpus). For all unsupervised systems, we choose the most frequent class if no counts are available. For SUMLM, we use the development sets to decide which orders of N-grams to combine, finding orders 3-5 optimal for preposition selection, 2-5 optimal for spelling correction, and 4-5 optimal for non-referential pronoun detection. Development experiments also showed RATIOLM works better starting from 4-grams, not the 5-grams originally used in [Carlson *et al.*, 2008].

5 Results

5.1 Preposition Selection

Preposition selection is a difficult task with a low baseline: choosing the most-common preposition (*of*) in our test set achieves 20.9%. Training on 7 million examples, Chodorow et al. [2007] achieved 69% on the full 34-way selection. Tetreault and Chodorow [2008] obtained a human upper bound by removing prepositions from text and asking annotators to fill in the blank with the best preposition (using the current sentence as context). Two annotators achieved only 75% agreement with each other and with the original text.

In light of these numbers, the accuracy of the N-gram models are especially impressive. SUPERLM reaches 75.4% accuracy, equal to the human agreement (but on different data). Performance continually improves with more training examples, but only by 0.25% from 300K to 1M examples (Figure 1). SUMLM (73.7%) significantly outperforms RATIOLM (69.7%), and nearly matches the performance of SUPERLM.TRIGRAM performs worst (58.8%), but note it is

	Max			
Min	2	3	4	5
2	50.2	63.8	70.4	72.6
3		66.8	72.1	73.7
4			69.3	70.6
5				57.8

Table 1: SUMLM accuracy (%) combining N-grams from order *Min* to *Max*

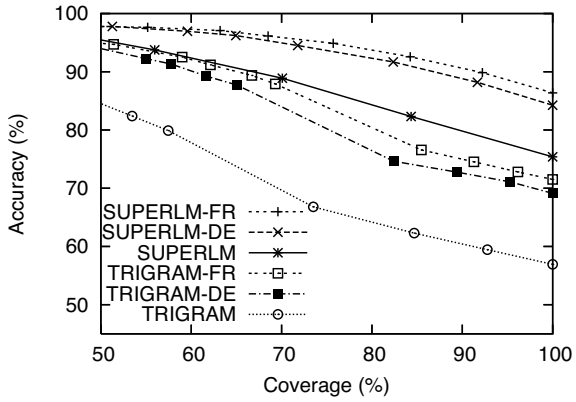

Figure 2: Preposition selection over high-confidence subsets, with and without language constraints (-FR,-DE)

the only previous web-scale approach applied to preposition selection [Felice and Pulman, 2007]. All differences are statistically significant (McNemar's test, $p<0.01$).

The order of N-grams used in the SUMLM system strongly affects performance. Using only trigrams achieves 66.8% accuracy, while using only 5-grams achieves just 57.8% (Table 1).[6] Summing counts from 3-5 results in the best performance on the development and test sets.

We compare our use of the Google Corpus to extracting page counts from a search engine, via the Google API. Since the number of queries allowed to the API is restricted, we test on only the first 1000 test examples. Using the Google Corpus, TRIGRAM achieves 61.1%, dropping to 58.5% with search engine page counts. Although this is a small difference, the real issue is the restricted number of queries allowed. For each example, SUMLM would need 14 counts for each of the 34 fillers instead of just one. For training SUPERLM, which has 1 million training examples, we need counts for 267 million *unique* N-grams. Using the Google API with a 1000-query-per-day quota, it would take over 732 years to collect all the counts for training. This is clearly why some web-scale systems use such limited context.

We also follow Carlson et al. [2001] and Chodorow et al. [2007] in extracting a subset of decisions where our system has higher confidence. We only propose a label if the ratio between the highest and second-highest score from our classifier is above a certain threshold, and then vary this threshold to produce accuracy at different coverage levels (Figure 2).

[6] Coverage is the main issue affecting the 5-gram model: only 70.1% of the test examples had a 5-gram count for *any* of the 34 fillers (93.4% for 4-grams, 99.7% for 3-grams)

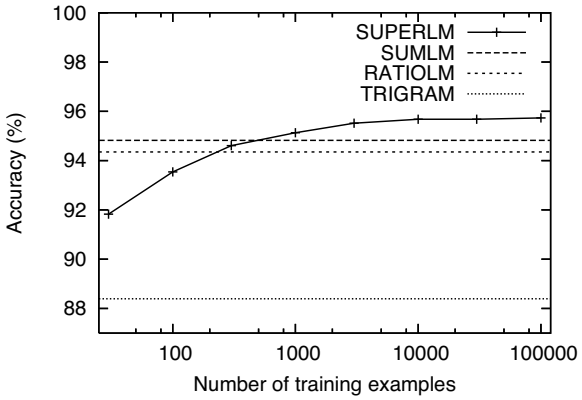

Figure 3: Context-sensitive spelling correction learning curve

The SUPERLM system can obtain close to 90% accuracy when deciding on 70% of examples, and above 95% accuracy when deciding on half the examples. The TRIGRAM performance rises more slowly as coverage drops, reaching 80% accuracy when deciding on only 57% of examples.

Many of SUPERLM's errors involve choosing between prepositions that are unlikely to be confused in practice, e.g. *with/without*. Chodorow et al. [2007] wrote post-processor rules to prohibit corrections in the case of antonyms. Note that the errors made by an English learner also depend on their native language. A French speaker looking to translate *au-dessus de* has one option in some dictionaries: *above*. A German speaker looking to translate *über* has, along with *above*, many more options. When making corrections, we could combine SUPERLM (a *source* model) with the likelihood of each confusion depending on the writer's native language (a *channel* model). This model could be trained on text written by second-language learners. In the absence of such data, we only allow our system to make corrections in English if the proposed replacement shares a foreign-language translation in a particular Freelang online bilingual dictionary.

To simulate the use of this module, we randomly flip 20% of our test-set prepositions to confusable ones, and then apply our classifier with the aforementioned confusability (and confidence) constraints. We experimented with French and German lexicons (Figure 2). These constraints strongly benefit both the SUPERLM and the TRIGRAM systems, with French constraints ($-FR$) helping slightly more than German ($-DE$) for higher coverage levels. There are fewer confusable prepositions in the French lexicon compared to German. As a baseline, if we assign our labels random scores, adding the French and German constraints results in 20% and 14% accuracy, respectively (compared to $\frac{1}{34}$ unconstrained). At 50% coverage, both constrained SUPERLM systems achieve close to 98% accuracy, a level that could provide very reliable feedback in second-language learning software.

5.2 Context-sensitive Spelling Correction

Figure 3 provides the spelling correction learning curve, while Table 2 gives results on the five confusion sets (Section 4.2). Choosing the most frequent label averages 66.9% on this task (BASE). TRIGRAM scores 88.4%, comparable

Set	BASE	TRIGRAM	SUMLM	SUPERLM
among	60.3	80.8	90.5	92.8
amount	75.6	83.9	93.2	93.7
cite	87.1	94.3	96.3	97.6
peace	60.8	92.3	97.7	98.0
raise	51.0	90.7	96.6	96.6
Avg.	66.9	88.4	94.8	95.7

Table 2: Context-sensitive spelling correction accuracy (%) on different confusion sets

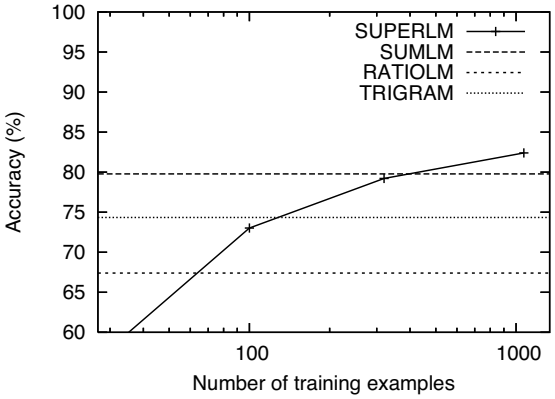

Figure 4: Non-referential detection learning curve

to the trigram (page count) results reported in [Lapata and Keller, 2005]. SUPERLM again achieves the highest performance (95.7%), and it reaches this performance using many fewer training examples than with preposition selection. This is because the number of parameters grows with the number of fillers *times* the number of labels, and there are 34 prepositions but only two-to-three confusable spellings.

SUPERLM achieves a 24% relative reduction in error over RATIOLM (94.4%), which was the previous state-of-the-art [Carlson *et al.*, 2008]. SUMLM (94.8%) also improves on RATIOLM, although results are generally similar on the different confusion sets. On {*raise,rise*}, SUPERLM's supervised weighting of the counts by position and size does not improve over SUMLM (Table 2). On all the other sets the performance is higher; for example, on {*among,between*}, the accuracy improves by 2.3%. On this set, counts for fillers near the beginning of the context pattern are more important, as the object of the preposition is crucial for distinguishing these two classes ("*between* the **two**" but "*among* the **three**"). SUPERLM can exploit the relative importance of the different positions and thereby achieve higher performance.

5.3 Non-referential Pronoun Detection

For non-referential pronoun detection, BASE (always choosing referential) achieves 59.4%, while SUPERLM reaches 82.4%. Bergsma et al. [2008] report state-of-the-art accuracy of 85.7%, over a baseline of 68.3%; thus in our data SUPERLM achieves a higher but similar relative reduction of error over BASE. RATIOLM, with no tuned thresholds, performs worst (67.4%), while TRIGRAM (74.3%) and SUMLM (79.8%) achieve reasonable performance by com-

paring scores for *it* and *they* (Section 4.3). All differences are statistically significant (McNemar's test, $p<0.05$), except between SUPERLM and SUMLM.

As this is our only task for which substantial effort was needed to create training data, we are particularly interested in the learning rate of SUPERLM (Figure 4). After 1070 examples, it does not yet show signs of plateauing. Here, SUPERLM uses double the number of fillers (hence double the parameters) that were used in spelling correction, and spelling performance did not level-off until after 10K training examples. Thus labeling an order of magnitude more data will likely also yield further improvements in SUPERLM.

However, note these efforts would have to be repeated in every new language and domain to which SUPERLM is applied. On the other hand, SUMLM performs almost as well as SUPERLM and requires no supervision. Furthermore, error analysis by Bergsma et al. [2008] indicates further gains in accuracy could come most easily by jointly optimizing detection with pronoun resolution. SUMLM would be a more competitive and convenient system for rapid development of systems that operate jointly over different languages and texts.

6 Conclusion

We presented a unified view of using web-scale N-gram models for lexical disambiguation. State-of-the-art results by our supervised and unsupervised systems demonstrate that it is not only important to use the largest corpus, but to get maximum information from this corpus. Using the Google 5-gram data not only provides better accuracy than using page counts from a search engine, but facilitates the use of more context of various sizes and positions. The TRIGRAM approach, popularized by Lapata and Keller [2005], clearly underperforms the unsupervised SUMLM system on all three applications.

In each of our tasks, the candidate set was pre-defined, and training data was available to train the supervised system. While SUPERLM achieves the highest performance, the simpler SUMLM system, which uses uniform weights, performs nearly as well as SUPERLM, and exceeds it for smaller training sizes. Unlike SUPERLM, SUMLM could easily be used in cases where the candidate sets are generated dynamically; for example, to assess the preceding-noun candidates for anaphora resolution.

References

[Banko and Brill, 2001] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *ACL*, pages 26–33, 2001.

[Bergsma *et al.*, 2008] Shane Bergsma, Dekang Lin, and Randy Goebel. Distributional identification of non-referential pronouns. In *ACL*, pages 10–18, 2008.

[Brants *et al.*, 2007] Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. Large language models in machine translation. In *EMNLP*, pages 858–867, 2007.

[Carlson *et al.*, 2001] Andrew J. Carlson, Jeffrey Rosen, and Dan Roth. Scaling up context-sensitive text correction. In *AAAI/IAAI*, pages 45–50, 2001.

[Carlson *et al.*, 2008] Andrew Carlson, Tom M. Mitchell, and Ian Fette. Data analysis project: Leveraging massive textual corpora using n-gram statistics. Technial Report CMU-ML-08-107, 2008.

[Chodorow *et al.*, 2007] Martin Chodorow, Joel R. Tetreault, and Na-Rae Han. Detection of grammatical errors involving prepositions. In *ACL-SIGSEM Workshop on Prepositions*, pages 25–30, 2007.

[Felice and Pulman, 2007] Rachele De Felice and Stephen G. Pulman. Automatically acquiring models of preposition use. In *ACL-SIGSEM Workshop on Prepositions*, pages 45–50, 2007.

[Gamon *et al.*, 2008] Michael Gamon, Jianfeng Gao, Chris Brockett, Alexandre Klementiev, William B. Dolan, Dmitriy Belenko, and Lucy Vanderwende. Using contextual speller techniques and language modeling for ESL error correction. In *IJCNLP*, 2008.

[Golding and Roth, 1999] Andrew R. Golding and Dan Roth. A Winnow-based approach to context-sensitive spelling correction. *Machine Learning*, 34(1-3):107–130, 1999.

[Kilgarriff, 2007] Adam Kilgarriff. Googleology is bad science. *Computational Linguistics*, 33(1):147–151, 2007.

[Lapata and Keller, 2005] Mirella Lapata and Frank Keller. Web-based models for natural language processing. *ACM Transactions on Speech and Language Processing*, 2(1):1–31, 2005.

[Litkowski and Hargraves, 2007] Ken Litkowski and Orin Hargraves. SemEval-2007 Task 06: Word-sense disambiguation of prepositions. In *SemEval*, pages 24–29, 2007.

[Liu and Curran, 2006] Vinci Liu and James R. Curran. Web text corpus for natural language processing. In *EACL*, pages 233–240, 2006.

[Mihalcea and Moldovan, 1999] Rada Mihalcea and Dan I. Moldovan. A method for word sense disambiguation of unrestricted text. In *ACL*, pages 152–158, 1999.

[Roth, 1998] Dan Roth. Learning to resolve natural language ambiguities: A unified approach. In *AAAI/IAAI*, pages 806–813, 1998.

[Tetreault and Chodorow, 2008] Joel R. Tetreault and Martin Chodorow. The ups and downs of preposition error detection in ESL writing. In *COLING*, 2008.

[Tsochantaridis *et al.*, 2004] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[Wilcox-O'Hearn *et al.*, 2008] Amber Wilcox-O'Hearn, Graeme Hirst, and Alexander Budanitsky. Real-word spelling correction with trigrams: A reconsideration of the Mays, Damerau, and Mercer model. In *CICLing*, pages 605–616, 2008.

[Yarowsky, 1994] David Yarowsky. Decision lists for lexical ambiguity resolution: application to accent restoration in spanish and french. In *ACL*, pages 88–95, 1994.