tRuEcasIng

Lucian Vlad Lita ♠
Carnegie Mellon
llita@cs.cmu.edu

Abe Ittycheriah
IBM T.J. Watson
abei@us.ibm.com

Salim Roukos
IBM T.J. Watson
roukos@us.ibm.com

Nanda Kambhatla
IBM T.J. Watson
nanda@us.ibm.com

Abstract

Truecasing is the process of restoring case information to badly-cased or non-cased text. This paper explores truecasing issues and proposes a statistical, language modeling based truecaser which achieves an accuracy of $\sim$98% on news articles. Task based evaluation shows a 26% F-measure improvement in named entity recognition when using truecasing. In the context of automatic content extraction, mention detection on automatic speech recognition text is also improved by a factor of 8. Truecasing also enhances machine translation output legibility and yields a BLEU score improvement of 80.2%. This paper argues for the use of truecasing as a valuable component in text processing applications.

1 Introduction

While it is true that large, high quality text corpora are becoming a reality, it is also true that the digital world is flooded with enormous collections of low quality natural language text. Transcripts from various audio sources, automatic speech recognition, optical character recognition, online messaging and gaming, email, and the web are just a few examples of raw text sources with content often produced in a hurry, containing misspellings, insertions, deletions, grammatical errors, neologisms, jargon terms etc. We want to enhance the quality of such sources in order to produce better rule-based systems and sharper statistical models.

This paper focuses on **truecasing**, which is the process of restoring case information to raw text. Besides text rEaDaBILiTY, truecasing enhances the quality of case-carrying data, brings into the picture new corpora originally considered too noisy for various NLP tasks, and performs case normalization across styles, sources, and genres.

Consider the following mildly ambiguous sentence "us rep. james pond showed up riding an it and going to a now meeting". The case-carrying alternative "US Rep. James Pond showed up riding an IT and going to a NOW meeting" is arguably better fit to be subjected to further processing.

Broadcast news transcripts contain casing errors which reduce the performance of tasks such as named entity tagging. Automatic speech recognition produces non-cased text. Headlines, teasers, section headers - which carry high information content - are not properly cased for tasks such as question answering. Truecasing is an essential step in transforming these types of data into cleaner sources to be used by NLP applications.

"the president" and "the President" are two viable surface forms that correctly convey the same information in the same context. Such discrepancies are usually due to differences in news source, authors, and stylistic choices. Truecasing can be used as a normalization tool across corpora in order to produce consistent, context sensitive, case information; it consistently reduces expressions to their statistical canonical form.

♠ Work done at IBM TJ Watson Research Center

In this paper, we attempt to show the benefits of truecasing in general as a valuable building block for NLP applications rather than promoting a specific implementation. We explore several truecasing issues and propose a statistical, language modeling based truecaser, showing its performance on news articles. Then, we present a straight forward application of truecasing on machine translation output. Finally, we demonstrate the considerable benefits of truecasing through task based evaluations on named entity tagging and automatic content extraction.

1.1 Related Work

Truecasing can be viewed in a lexical ambiguity resolution framework (Yarowsky, 1994) as discriminating among several versions of a word, which happen to have different surface forms (casings). Word-sense disambiguation is a broad scope problem that has been tackled with fairly good results generally due to the fact that context is a very good predictor when choosing the sense of a word. (Gale et al., 1994) mention good results on limited case restoration experiments on toy problems with 100 words. They also observe that real world problems generally exhibit around 90% case restoration accuracy. (Mikheev, 1999) also approaches casing disambiguation but models only instances when capitalization is expected: first word in a sentence, after a period, and after quotes. (Chieu and Ng, 2002) attempted to extract named entities from non-cased text by using a weaker classifier but without focusing on regular text or case restoration.

Accents can be viewed as additional surface forms or alternate word casings. From this perspective, either accent identification can be extended to truecasing or truecasing can be extended to incorporate accent restoration. (Yarowsky, 1994) reports good results with statistical methods for Spanish and French accent restoration.

Truecasing is also a specialized method for spelling correction by relaxing the notion of casing to spelling variations. There is a vast literature on spelling correction (Jones and Martin, 1997; Golding and Roth, 1996) using both linguistic and statistical approaches. Also, (Brill and Moore, 2000) apply a noisy channel model, based on generic string to string edits, to spelling correction.

2 Approach

In this paper we take a statistical approach to truecasing. First we present the baseline: a simple, straight forward unigram model which performs reasonably well in most cases. Then, we propose a better, more flexible statistical truecaser based on language modeling.

From a truecasing perspective we observe four general classes of words: all lowercase (*LC*), first letter uppercase (*UC*), all letters uppercase (*CA*), and mixed case word *MC*). The *MC* class could be further refined into meaningful subclasses but for the purpose of this paper it is sufficient to correctly identify specific true *MC* forms for each *MC* instance.

We are interested in correctly assigning case labels to words (tokens) in natural language text. This represents the ability to discriminate between class labels for the same lexical item, taking into account the surrounding words. We are interested in casing word combinations observed during training as well as new phrases. The model requires the ability to generalize in order to recognize that even though the possibly misspelled token "lenon" has never been seen before, words in the same context usually take the *UC* form.

2.1 Baseline: The Unigram Model

The goal of this paper is to show the benefits of truecasing in general. The unigram baseline (presented below) is introduced in order to put task based evaluations in perspective and **not** to be used as a strawman baseline.

The vast majority of vocabulary items have only one surface form. Hence, it is only natural to adopt the unigram model as a baseline for truecasing. In most situations, the unigram model is a simple and efficient model for surface form restoration. This method associates with each surface form a score based on the frequency of occurrence. The decoding is very simple: the *true* case of a token is predicted by the most likely case of that token.

The unigram model's upper bound on truecasing performance is given by the percentage of tokens that occur during decoding under their most frequent case. Approximately 12% of the vocabulary items have been observed under more than one surface form. Hence it is inevitable for the unigram model to fail on tokens such as "new". Due to the overwhelming frequency of its *LC* form, "new" will take this particular form regardless of what token follows it. For both "information" and "york" as subsequent words, "new" will be labeled as *LC*. For the latter case, "new" occurs under one of its less frequent surface forms.

2.2 Truecaser

The truecasing strategy that we are proposing seeks to capture local context and bootstrap it across a sentence. The case of a token will depend on the most likely meaning of the sentence - where local meaning is approximated by n-grams observed during training. However, the local context of a few words alone is not enough for case disambiguation. Our proposed method employs sentence level context as well.

We capture local context through a trigram language model, but the case label is decided at a sentence level. A reasonable improvement over the unigram model would have been to decide the word casing given the previous two lexical items and their corresponding case content. However, this greedy approach still disregards global cues. Our goal is to maximize the probability of a larger text segment (i.e. a sentence) occurring under a certain surface form. Towards this goal, we first build a language model that can provide local context statistics.

2.2.1 Building a Language Model

Language modeling provides features for a labeling scheme. These features are based on the probability of a lexical item and a case content conditioned on the history of previous two lexical items and their corresponding case content:

$$\begin{aligned} P_{model}(w_3|w_2,w_1) &= \lambda_{trigram}P(w_3|w_2,w_1) \\ &+ \lambda_{bigram}P(w_3|w_2) \\ &+ \lambda_{unigram}P(w_3) \\ &+ \lambda_{uniform}P_0 \qquad (1) \end{aligned}$$

where trigram, bigram, unigram, and uniform probabilities are scaled by individual λ_is which are learned by observing training examples. w_i represents a word with a case tag treated as a unit for probability estimation.

2.2.2 Sentence Level Decoding

Using the language model probabilities we decode the case information at a sentence level. We construct a trellis (figure 1) which incorporates all the sentence surface forms as well as the features computed during training. A node in this trellis consists of a lexical item, a position in the sentence, a possible casing, as well as a history of the previous two lexical items and their corresponding case content. Hence, for each token, all surface forms will appear as nodes carrying additional context information. In the trellis, thicker arrows indicate higher transition probabilities.

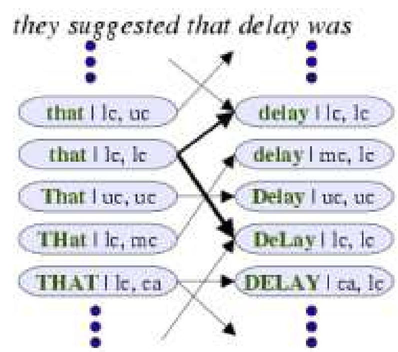

Figure 1: Given individual histories, the decodings *delay* and *DeLay*, are most probable - perhaps in the context of "time delay" and respectively "Senator Tom DeLay"

The trellis can be viewed as a Hidden Markov Model (HMM) computing the state sequence which best explains the observations. The states $(q_1, q_2, \cdots, q_n)$ of the HMM are combinations of case and context information, the transition probabilities are the language model (λ) based features, and the observations $(O_1 O_2 \cdots O_t)$ are lexical items. During decoding, the Viterbi algorithm (Rabiner, 1989) is used to compute the highest probability state sequence (q_τ^* at sentence level) that yields the desired case information:

$$q_\tau^* = argmax_{q_{i1}q_{i2}\cdots q_{it}} P(q_{i1}q_{i2}\cdots q_{it}|O_1 O_2 \cdots O_t, \lambda) \qquad (2)$$

where $P(q_{i1}q_{i2}\cdots q_{it}|O_1 O_2 \cdots O_t, \lambda)$ is the probability of a given sequence conditioned on the observation sequence and the model parameters. A more sophisticated approach could be envisioned, where either the observations or the states are more expressive. These alternate design choices are not explored in this paper.

Testing speed depends on the width and length of the trellis and the overall decoding complexity is: $C_{decoding} = O(SM^{H+1})$ where S is the sentence size, M is the number of surface forms we are willing to consider for each word, and H is the history size ($H = 3$ in the trigram case).

2.3 Unknown Words

In order for truecasing to be generalizable it must deal with unknown words — words not seen during training. For large training sets, an extreme assumption is that most words and corresponding casings possible in a language have been observed during training. Hence, most new tokens seen during decoding are going to be either proper nouns or misspellings. The simplest strategy is to consider all unknown words as being of the *UC* form (i.e. people's names, places, organizations).

Another approach is to replace the less frequent vocabulary items with case-carrying special tokens. During training, the word *mispeling* is replaced with by *UNKNOWN_LC* and the word *Lenon* with *UNKNOWN_UC*. This transformation is based on the observation that similar types of infrequent words will occur during decoding. This transformation creates the precedent of unknown words of a particular format being observed in a certain context. When a truly unknown word will be seen in the same context, the most appropriate casing will be applied. This was the method used in our experiments. A similar method is to apply the case-carrying special token transformation only to a small random sample of **all** tokens, thus capturing context regardless of frequency of occurrence.

2.4 Mixed Casing

A reasonable truecasing strategy is to focus on token classification into three categories: *LC*, *UC*, and *CA*. In most text corpora mixed case tokens such as McCartney, CoOl, and TheBeatles occur with moderate frequency. Some NLP tasks might prefer mapping *MC* tokens starting with an uppercase letter into the *UC* surface form. This technique will reduce the feature space and allow for sharper models. However, the decoding process can be generalized to include mixed cases in order to find a closer fit to the true sentence. In a clean version of the AQUAINT (ARDA) news stories corpus, $\sim 90\%$ of the tokens occurred under the most frequent surface form (figure 2).

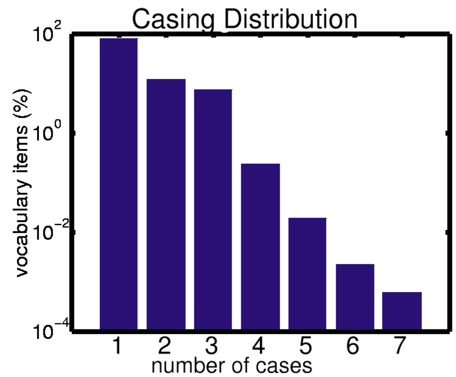

Figure 2: News domain casing distribution

The expensive brute force approach will consider all possible casings of a word. Even with the full casing space covered, some mixed cases will not be seen during training and the language model probabilities for n-grams containing certain words will back off to an unknown word strategy. A more feasible method is to account only for the mixed case items observed during training, relying on a large enough training corpus. A variable beam decoding will assign non-zero probabilities to all known casings of each word. An n-best approximation is somewhat faster and easier to implement and is the approach employed in our experiments. During the sentence-level decoding only the n-most-frequent mixed casings seen during training are considered. If the true capitalization is not among these n-best versions, the decoding is not correct. Additional lexical and morphological features might be needed if identifying *MC* instances is critical.

2.5 First Word in the Sentence

The first word in a sentence is generally under the UC form. This sentence-begin indicator is sometimes ambiguous even when paired with sentence-end indicators such as the period. While sentence splitting is not within the scope of this paper, we want to emphasize the fact that many NLP tasks would benefit from knowing the true case of the first word in the sentence, thus avoiding having to learn the fact that beginning of sentences are artificially important. Since it is uneventful to convert the first letter of a sentence to uppercase, a more interesting problem from a truecasing perspective is to learn how to predict the correct case of the first word in a sentence (i.e. not always *UC*).

If the language model is built on clean sentences accounting for sentence boundaries, the decoding will most likely uppercase the first letter of any sentence. On the other hand, if the language model is trained on clean sentences disregarding sentence boundaries, the model will be less accurate since different casings will be presented for the same context and artificial n-grams will be seen when transitioning between sentences. One way to obtain the desired effect is to discard the first n tokens in the training sentences in order to escape the sentence-begin effect. The language model is then built on smoother context. A similar effect can be obtained by initializing the decoding with n-gram state probabilities so that the boundary information is masked.

3 Evaluation

Both the unigram model and the language model based truecaser were trained on the AQUAINT (ARDA) and TREC (NIST) corpora, each consisting of 500M token news stories from various news agencies. The truecaser was built using IBM's ViaVoice™ language modeling tools. These tools implement trigram language models using deleted interpolation for backing off if the trigram is not found in the training data. The resulting model's perplexity is 108.

Since there is no absolute truth when truecasing a sentence, the experiments need to be built with some reference in mind. Our assumption is that professionally written news articles are very close to an intangible absolute truth in terms of casing. Furthermore, we ignore the impact of diverging stylistic forms, assuming the differences are minor.

Based on the above assumptions we judge the truecasing methods on four different test sets. The first test set (APR) consists of the August 25, 2002 * top 20 news stories from Associated Press and Reuters excluding titles, headlines, and section headers which together form the second test set (APR+). The third test set (ACE) consists of earlier news stories from AP and New York Times belonging to the ACE dataset. The last test set (MT) includes a set of machine translation references (i.e. human translations) of news articles from the Xinhua agency. The sizes of the data sets are as follows: APR - 12k tokens, ACE - 90k tokens, and MT - 63k tokens. For both truecasing methods, we computed the agreement with the original news story considered to be the ground truth.

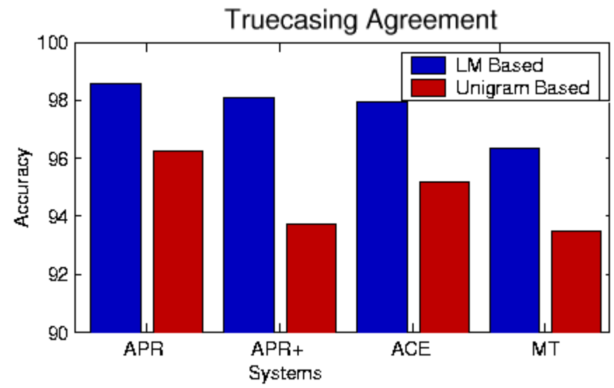

Figure 3: LM truecaser vs. unigram baseline.

3.1 Results

The language model based truecaser consistently displayed a significant error reduction in case restoration over the unigram model (figure 3). On current news stories, the truecaser agreement with the original articles is $\sim 98\%$.

Titles and headlines usually have a higher concentration of named entities than normal text. This also means that they need a more complex model to assign case information more accurately. The LM based truecaser performs better in this environment while the unigram model misses named entity components which happen to have a less frequent surface form.

3.2 Qualitative Analysis

The original reference articles are assumed to have the absolute true form. However, differences from these original articles and the truecased articles are not always casing errors. The truecaser tends to modify the first word in a quotation if it is not proper name: "There has been" becomes "there has been". It also makes changes which could be considered a correction of the original article: "Xinhua

*Randomly chosen test date

		BLEU Breakdown			
System	BLEU	1gr Precision	2gr Precision	3gr Precision	4gr Precision
all lowercase	0.1306	0.6016	0.2294	0.1040	0.0528
rule based	0.1466	0.6176	0.2479	0.1169	0.0627
1gr truecasing	0.2206	0.6948	0.3328	0.1722	0.0988
1gr truecasing+	0.2261	0.6963	0.3372	0.1734	0.0997
lm truecasing	0.2596	0.7102	0.3635	0.2066	0.1303
lm truecasing+	**0.2642**	**0.7107**	**0.3667**	**0.2066**	**0.1302**

Table 1: BLEU score for several truecasing strategies. (*truecasing+* methods additionally employ the "first sentence letter uppercased" rule adjustment).

	Baseline			With Truecasing		
Class	Recall	Precision	**F**	Recall	Precision	F
ENAMEX	48.46	36.04	**41.34**	59.02	52.65	**55.66** (+34.64%)
NUMEX	64.61	72.02	**68.11**	70.37	79.51	**74.66** (+9.62%)
TIMEX	47.68	52.26	**49.87**	61.98	75.99	**68.27** (+36.90%)
Overall	52.50	44.84	**48.37**	62.01	60.42	**61.20** (+26.52%)

Table 2: Named Entity Recognition performance with truecasing and without (baseline).

news agency" becomes "Xinhua News Agency" and "northern alliance" is truecased as "Northern Alliance". In more ambiguous cases both the original version and the truecased fragment represent different stylistic forms: "prime minister Hekmatyar" becomes "Prime Minister Hekmatyar".

There are also cases where the truecaser described in this paper makes errors. New movie names are sometimes miss-cased: "my big fat greek wedding" or "signs". In conducive contexts, person names are correctly cased: "DeLay said in". However, in ambiguous, adverse contexts they are considered to be common nouns: "pond" or "to delay that". Unseen organization names which make perfectly normal phrases are erroneously cased as well: "international security assistance force".

3.3 Application: Machine Translation Post-Processing

We have applied truecasing as a post-processing step to a state of the art machine translation system in order to improve readability. For translation between Chinese and English, or Japanese and English, there is no transfer of case information. In these situations the translation output has no case information and it is beneficial to apply truecasing as a post-processing step. This makes the output more legible and the system performance increases if case information is required.

We have applied truecasing to Chinese-to-English translation output. The data source consists of news stories (2500 sentences) from the Xinhua News Agency. The news stories are first translated, then subjected to truecasing. The translation output is evaluated with BLEU (Papineni et al., 2001), which is a robust, language independent automatic machine translation evaluation method. BLEU scores are highly correlated to human judges scores, providing a way to perform frequent and accurate automated evaluations. BLEU uses a modified n-gram precision metric and a weighting scheme that places more emphasis on longer n-grams.

In table 1, both truecasing methods are applied to machine translation output with and without uppercasing the first letter in each sentence. The truecasing methods are compared against the all letters lowercased version of the articles as well as against an existing rule-based system which is aware of a limited number of entity casings such as dates, cities, and countries. The LM based truecaser is very effective in increasing the readability of articles and captures an important aspect that the BLEU score is sensitive to. Truecasig the translation output yields

	Baseline			With Truecasing		
Source	Recall	Precision	F	Recall	Precision	F
BNEWS ASR	23	3	**5**	56	39	**46** (+820.00%)
BNEWS HUMAN	77	66	**71**	77	68	**72** (+1.41%)
XINHUA	76	71	**73**	79	72	**75** (+2.74%)

Table 3: Results of ACE mention detection with and without truecasing.

an improvement † of 80.2% in BLEU score over the existing rule base system.

3.4 Task Based Evaluation

Case restoration and normalization can be employed for more complex tasks. We have successfully leveraged truecasing in improving named entity recognition and automatic content extraction.

3.4.1 Named Entity Tagging

In order to evaluate the effect of truecasing on extracting named entity labels, we tested an existing named entity system on a test set that has significant case mismatch to the training of the system. The base system is an HMM based tagger, similar to (Bikel et al., 1997). The system has 31 semantic categories which are extensions on the MUC categories. The tagger creates a lattice of decisions corresponding to tokenized words in the input stream. When tagging a word w_i in a sentence of words $w_0 ... w_N$, two possibilities. If a tag begins:

$$p(t_1^N | w_1^N)_i = p(t_i | t_{i-1}, w_{i-1}) p^\dagger(w_i | t_i, w_{i-1})$$

If a tag continues:

$$p(t_1^N | w_1^N)_i = p(w_i | t_i, w_{i-1})$$

The † indicates that the distribution is formed from words that are the first words of entities. The $p^\dagger$ distribution predicts the probability of seeing that word given the tag and the previous word instead of the tag and previous tag. Each word has a set of features, some of which indicate the casing and embedded punctuation. These models have several levels of back-off when the exact trigram has not been seen in training. A trellis spanning the 31 futures is built for each word in a sentence and the best path is derived using the Viterbi algorithm.

† Truecasing improves legibility, not the translation itself

The performance of the system shown in table 2 indicate an overall 26.52% F-measure improvement when using truecasing. The alternative to truecasing text is to destroy case information in the training material ⊖ SNORIFY procedure in (Bikel et al., 1997). Case is an important feature in detecting most named entities but particularly so for the title of a work, an organization, or an ambiguous word with two frequent cases. Truecasing the sentence is essential in detecting that “To Kill a Mockingbird” is the name of a book, especially if the quotation marks are left off.

3.4.2 Automatic Content Extraction

Automatic Content Extraction (ACE) is task focusing on the extraction of mentions of entities and relations between them from textual data. The textual documents are from newswire, broadcast news with text derived from automatic speech recognition (ASR), and newspaper with text derived from optical character recognition (OCR) sources. The mention detection task (ace, 2001) comprises the extraction of named (e.g. ”Mr. Isaac Asimov”), nominal (e.g. ”the complete author”), and pronominal (e.g. ”him”) mentions of Persons, Organizations, Locations, Facilities, and Geo-Political Entities.

The automatically transcribed (using ASR) broadcast news documents and the translated Xinhua News Agency (XINHUA) documents in the ACE corpus do not contain any case information, while human transcribed broadcast news documents contain casing errors (e.g. “George bush”). This problem occurs especially when the data source is noisy or the articles are poorly written.

For all documents from broadcast news (human transcribed and automatically transcribed) and XINHUA sources, we extracted mentions before and after applying truecasing. The ASR transcribed broadcast news data comprised 86 documents containing a total of 15,535 words, the human transcribed version contained 15,131 words. There were only two XINHUA documents in the ACE test set containing a total of 601 words. None of this data or any ACE data was used for training the truecasing models.

Table 3 shows the result of running our ACE participating maximum entropy mention detection system on the raw text, as well as on truecased text. For ASR transcribed documents, we obtained an eight fold improvement in mention detection from 5% F-measure to 46% F-measure. The low baseline score is mostly due to the fact that our system has been trained on newswire stories available from previous ACE evaluations, while the latest test data included ASR output. It is very likely that the improvement due to truecasing will be more modest for the next ACE evaluation when our system will be trained on ASR output as well.

4 Possible Improvements & Future Work

Although the statistical model we have considered performs very well, further improvements must go beyond language modeling, enhancing how expressive the model is. Additional features are needed during decoding to capture context outside of the current lexical item, medium range context, as well as discontinuous context. Another potentially helpful feature to consider would provide a distribution over similar lexical items, perhaps using an edit/phonetic distance.

Truecasing can be extended to cover a more general notion surface form to include accents. Depending on the context, words might take different surface forms. Since punctuation is a notion extension to surface form, shallow punctuation restoration (e.g. *word followed by comma*) can also be addressed through truecasing.

5 Conclusions

We have discussed truecasing, the process of restoring case information to badly-cased or non-cased text, and we have proposed a statistical, language modeling based truecaser which has an agreement of $\sim$98% with professionally written news articles. Although its most direct impact is improving legibility, truecasing is useful in case normalization across styles, genres, and sources. Truecasing is a valuable component in further natural language processing. Task based evaluation shows a 26% F-measure improvement in named entity recognition when using truecasing. In the context of automatic content extraction, mention detection on automatic speech recognition text is improved by a factor of 8. Truecasing also enhances machine translation output legibility and yields a BLEU score improvement of 80.2% over the original system.

References

2001. Entity detection and tracking. *ACE Pilot Study Task Definition*.

D. Bikel, S. Miller, R. Schwartz, and R. Weischedel. 1997. Nymble: A high-performance learning name finder. pages 194–201.

E. Brill and R. C. Moore. 2000. An improved error model for noisy channel spelling correction. *ACL*.

H.L. Chieu and H.T. Ng. 2002. Teaching a weaker classifier: Named entity recognition on upper case text.

William A. Gale, Kenneth W. Church, and David Yarowsky. 1994. Discrimination decisions for 100,000-dimensional spaces. *Current Issues in Computational Linguistics*, pages 429–450.

Andrew R. Golding and Dan Roth. 1996. Applying winnow to context-sensitive spelling correction. *ICML*.

M. P. Jones and J. H. Martin. 1997. Contextual spelling correction using latent semantic analysis. *ANLP*.

A. Mikheev. 1999. A knowledge-free method for capitalized word disambiguation.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. *IBM Research Report*.

L. R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Readings in Speech Recognition*, pages 267–295.

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