CS 674 project final report David Crandall

Automatic accent restoration in Spanish text

1 Introduction

Written accents (diacritics) over vowels in Spanish are important cues to word meaning and pronunciation. However, in informal writing (such as e-mail messages) it is common for writers to omit the accents. In all but the most pathological of cases, a native reader can still infer the writer's intended meaning using context cues. However, missing accents can still confuse or slow a reader, especially one who is not a native speaker.

An automatic accent restoration algorithm is an attractive solution to this problem. Given Spanish text in which written accents have been omitted, the algorithm would add accents, as necessary, based on context to preserve the writer's meaning. Such an algorithm would allow the writer to avoid the extra hassle of typing the diacritics, but allow the reader to still enjoy properly-accented text. Such an algorithm could potentially even be applied in real-time by a writer's e-mail software or word processor.

This paper explores the use of statistical algorithms to automatically restore accents in Spanish text. After a brief introduction to the role of written accents in Spanish and a summary of related work, we describe three algorithms that we have applied to the accent restoration task. First, we describe a Bayesian framework that uses surrounding words to make accent restoration decisions, viewing the problem like a confusion set disambiguation task. Second, we describe a Hidden Markov Model-based algorithm that uses part-of-speech tagging to infer correct accentuation patterns. Lastly, we combine the Bayesian and HMM methods to produce a hybrid method that outperforms either individual approach. In Section 3 we present extensive experimentation evaluating the various approaches and comparing them to prior work in the literature.

A key question we study in this paper is how the quality and size of the training corpus affects the accuracy of NLP methods on the accent restoration task. While professional Spanish corpora are commercially available, we chose instead to collect our own corpus by automatically crawling Spanish-language websites. Collecting a corpus in this way is extremely cheap, but potentially introduces pollution (such as foreign words and incorrectly-accented words) into the training corpus. We believe that the results of our experiments may be interesting to researchers interested not only in accent restoration, but also to those using Bayesian- and HMM-based frameworks for other NLP problems.

1.1 Background: The role of diacritics in Spanish

As in many European languages, Spanish has diacritic marks that can be placed on certain letters. There are three diacritic marks in Spanish: the acute accent that may be placed over vowels (as in *dé*), the tilde that may be placed over the letter *n* (as in *año*), and the diaeresis mark that may be placed over the letter *u* (as in *ambigüedad*). However, the latter marks are rare[1] and can usually be resolved using a lexicon alone (i.e. leaving off these marks does not usually introduce ambiguity). In this paper we concentrate on the most interesting diacritics from an NLP perspective: the acute accents on vowels.

For background, we briefly describe the role of written accents in Spanish; more detail can be found in [5]. Accented vowels in Spanish serve primarily as an aid to pronunciation. However, for some words, adding or removing an accent changes their meaning. We can therefore classify a given Spanish word into one of the following categories:

1. **Words without ambiguity:** For these words, there is only one possible correct accentuation pattern. For example, the word *acentuación* always has an accent over the *o*, and *acentuacion* does not exist in the language.

[1] In our corpus, the tilde occurs about once every 150 words. The diaeresis mark is exceptionally rare, occurring only about once every 6,000 words. In contrast, acute accents appear in about one of every ten words.

2. **Words with ambiguity:** For other words, adding or removing a written accent changes the meaning. That is, given a word written (correctly or incorrectly) without an accent, there is an ambiguity set containing the possible accentuation patterns. We can further classify the types of ambiguity sets:

   (a) **Different conjugations of the same verb:** In this case, the possible accentuations are all verbs, and the written accent differentiates between verb tenses, moods, and person. Examples include: *canto* (I sing) and *cantó* (he/she sang), *tomaras* (you took, past subjunctive) and *tomarás* (you will take), and *hable* (he/she speaks, subjunctive) and *hablé* (I spoke).

   (b) **Different parts of speech:** Some examples are: *de* (preposition, *of*) and *dé* (imperative verb, *give*), *hacia* (preposition, *toward*) and *hacía* (verb, *he/she/I did*), and *si* (conjunction, *if*) and *sí* (adverb, *yes*).

   (c) **Same part of speech (but not conjugated verbs):** For example, *papa* and *papá* are both nouns, meaning *potato* and *father*, respectively. A *secretaria* is a secretary in an office, while a *secretaría* is a government position (e.g. Secretary of State).

These observations on the role of written accents in Spanish have consequences for an accent restoration algorithm. For example, type 1 words can be processed in isolation, without any contextual information. These words could be handled by looking up the correct accentuation pattern in a lexicon. Restoring accents in type 2 words requires examining the word's context. For type 2(a) words, this usually involves identifying the subject of the verb. However, as in the *tomaras/tomarás* example above, in some cases it is also necessary to infer the correct tense and/or mood. Inferring the subject of a verb is particularly difficult in Spanish because subject pronouns are usually optional. That is, while "*Sang.*" is not a valid English sentence because there is no subject, "*Cantó.*" is a perfectly valid sentence in Spanish, but it is impossible to determine if the subject is a person, a thing, or a formal second person without examining surrounding sentences. Type 2(b) words can be disambiguated by choosing the word having the correct part of speech, as hypothesized by an automatic part-of-speech tagger, for example. Type 2(c) words require a higher-level semantic analysis of the surrounding text to decide which accentuation pattern best fits the context.

1.2 Related work

We presented a detailed literature survey of work related to the accent restoration problem in [2]. We very briefly summarize this survey here. Yarowsky [10][9] applies Bayesian classifiers, HMMs, and decision lists to accent restoration in French and Spanish. The methods are evaluated against a large corpus, but for only a few sets of ambiguous words. Simard & Deslauriers [7] restores accents in French by modeling parts of speech using an HMM. Their method requires a large corpus marked with parts of speech for training, and is unable to resolve ambiguities among the same parts of speech. Bolshakov et al [6][1] take a rule-based approach to restoring accents in Spanish. They resolve ambiguities between nouns and verbs by examining immediately surrounding words, taking advantage of Spanish's noun-adjective agreement rules for gender and plurality. Their system is unable to resolve ambiguities among candidate words with the same parts of speech. Mihalcea & Nastasemihalcea [8] model diacritic marks at the individual letter level. This is useful when a lexicon for a target language is unavailable, but because they process individual words at a time, their approach cannot perform disambiguation among multiple possible accentuation patterns.

2 Our approach

In this section, we present three methods for accent restoration: a Bayesian framework, an HMM framework, and a hybrid approach incorporating both methods. These methods are used for the experiments reported in Section 3.

2.1 Bayesian framework

The idea behind our Bayesian framework is that words occurring near an ambiguous stripped word will give clues to the correct accentuation pattern. For example, when choosing between *papa* (*potato*) and *papá* (*father*), the occurrence of words related to food or family may help to make the decision. A very similar approach was applied to accent restoration in [10].

Suppose that a stripped word w_0 in the test corpus has multiple possible accent patterns $a_0, a_1, ...a_n$. The task is then to choose the most likely accent pattern given the words surrounding w_0 in the text. To do this, we first choose a window size, specifying the number of words surrounding w_0 that will be examined. Suppose we choose a window size of s words on either side of w_0. Then the probability of a given accent pattern a_i given the surrounding context can be written as $P(a_i|w_{-s}, ..., w_{-1}, w_1, ..., w_s)$. The task is then to choose the accent pattern that maximizes this probability; i.e. choose $i*$ such that

$$i* = \text{argmax}_{0 \leq i \leq n} P(a_i|w_{-s}, ..., w_{-1}, w_1, ..., w_s) \tag{1}$$

By Baye's rule, the conditional probability of a_i can be rewritten as:

$$P(a_i|w_{-s}, ..., w_{-1}, w_1, ..., w_s) = \frac{P(w_{-s}, ..., w_{-1}, w_1, ..., w_s|a_i)P(a_i)}{P(w_{-s}, ..., w_{-1}, w_1, ..., w_s)} \tag{2}$$

We model the context as an unordered bag of independent words. Under this assumption, equation 2 can be rewritten as:

$$P(a_i|w_{-s}, ..., w_{-1}, w_1, ..., w_s) = P(a_i) \prod_{1 \leq |j| \leq s} \frac{P(w_j|a_i)}{P(w_j)} \tag{3}$$

where $P(w_j|a_i)$ is the probability that a word w_j appears within the context of the word w_0 with accent pattern a_i. Since the denominator is independent of the accent pattern being considered, we can rewrite equation 1 as:

$$i* = \text{argmax}_{0 \leq i \leq n} P(a_i) \prod_{1 \leq |j| \leq s} P(w_j|a_i) \tag{4}$$

The probabilities $P(w_j|a_i)$ and $P(a_i)$ can be directly estimated from the training data. We use a very simple smoothing technique that sets any zero probabilities to a very small but non-zero constant. The window size parameter must also be specified. In Section 3, we experiment with different values for the window size, and also explore an alternative approach that uses different, optimal window sizes for each word.

2.2 HMM method

As mentioned in Section 1.1, some ambiguity sets in Spanish consist of words with different parts of speech. If the expected part of a speech of a given word could be predicted based on context, then the accent restoration problem could be solved by simply choosing the accentuation pattern having that part of speech.

We use Hidden Markov Models to perform part-of-speech tagging for Spanish accent restoration using a bi-gram model. HMMs have been used extensively for part-of-speech tagging before (see [3], for example). We could train a full part-of-speech tagger for Spanish by training the HMM on a corpus hand-labeled with part-of-speech tags. This is the approach Simard et al [7] used for accent restoration in French. However, manually labeling a training corpus with part-of-speech tags is extremely time consuming and requires a deep linguistic knowledge of the target language.

Instead, we used an approach inspired by the work of Yarowsky [10]. He notes that the morphology of Spanish is rich enough that part-of-speech can often be inferred by word suffixes alone. For example, words ending in *mente* are almost always adverbs, and words ending in *ar*, *er*, and *ir* are almost always verb infinitives. Of course, this technique is not perfect: a word ending in *o* may be a masculine noun, an adjective, or a conjugated verb in the first-person present tense. However, we have found that some simple pattern-matching rules can classify most words into pseudo-part-of-speech categories.

We used our pattern-matching rules to generate the labeled training corpus for the HMM training procedure. Then, to apply the HMM model to a test corpus, we performed Viterbi decoding on a per-sentence basis, where the set of possible labels for each stripped word is composed of the different part-of-speech categories that the word could be assigned to, depending on accent pattern. Once decoding is complete, the accent pattern corresponding to the category assigned to an ambiguous word by the HMM is selected.

An example of how this process works in practice is in order. Suppose we have already trained our HMM model, and we wish to apply it to the simple stripped sentence *Ella canto.* There is ambiguity in the second word, because the writer may have meant either *canto* (*I sing*) or *cantó* (*he/she sang*). To a Spanish speaker, the answer is obvious: *ella* is the pronoun *she*, so obviously the second accentuation pattern is correct. However for illustration we show how HMM decoding arrives at this result. *Ella* is assigned to a category representing third-person pronouns by our morphological analysis technique. The analysis concludes that *canto* can either belong to the class of first person present verbs (if it ends in *o*) or the class of third person past tense verbs (if it ends in *ó*). HMM decoding assigns it to the latter class, however, since a transition from a third-person pronoun to a first-person conjugated verb is highly unlikely. The restoration algorithm concludes that *cantó* is therefore the correct form.

2.3 Hybrid approach

The Bayesian framework and the HMM method each have various advantages and disadvantages. The accuracy of the HMM method is limited by the imperfect morphological analysis, and it is useless for ambiguity sets where all words correspond to the same part of speech. On the other hand, the Bayesian framework performs poorly for infrequent words for which insufficient training data exists, and is not capable of incorporating word order into its decisions.

It is therefore logical to consider using a hybrid approach that leverages the strengths of both the Bayesian and HMM methods. We accomplish this by choosing the method that works best model for each ambiguity set. That is, during training we can train both the HMM and Bayesian methods, and then test the models on the *training* data. The performance of each model is computed on a per-stripped word (per-ambiguity set) basis, and the best for each word is noted. Then when applying the method to a test corpus, we alternate back and forth between the two models, selecting the best model according to the word under consideration.

3 Experimental results

3.1 Corpus for testing and training

For training and testing purposes, we required a large corpus of Spanish text, ideally with perfect orthography. While other researchers have assembled Spanish corpora (e.g. [4]), we were unable to find a freely-available corpus that was large and diverse enough for our purposes. We therefore assembled our own corpus of Spanish text by crawling Spanish-language websites from the Internet. We tried to ensure that only pages with high-quality orthography were included by only crawling high-quality, professional sites of major newspapers, government offices, religious institutions, etc.

The corpus was obtained as follows. Good candidate websites for collecting high-quality Spanish text were identified by hand. The wget utility was used to crawl those websites and lynx was used to render the HTML files to ASCII text. The text files were then passed through a filter custom-built for each website that stripped out headers, footers, and menus.[2] A series of filters were then applied to each article. Punctuation marks were converted into special word tokens (e.g. periods were replaced with the token _period_). Numbers that appeared to be years (i.e. in the range 1900 to 2099) were replaced with the special token _year_. All remaining numbers were converted to the token _number_.

Two scripts were then applied to the resulting corpus to try improve its orthographic quality. First, documents having accents on less than 3% of words were deleted from the corpus. Most of these documents either did not have good orthography (i.e. many accents were missing) or were written in English. A second script attempted to remove portions of texts that were in languages other than Spanish. This was accomplished by manually collecting a few common words in English, French and Portuguese that are not words in Spanish and then executing a script that automatically removed paragraphs of text containing these

[2] It is desirable to remove headers, footers, and menus because they skew word frequency statistics by appearing in all documents.

# documents	# words (millions)	Source URL	Description
14817	8.9	http://www.elsalvador.com/	newspaper articles from El Diario de Hoy (San Salvador, El Salvador)
7163	2.8	http://enciclopedia.us.es/	encyclopedia articles from the Free Universal Encyclopedia in Spanish project
3593	1.7	http://www.umm.edu/esp_ency/	articles from the University of Maryland Medical Center's Spanish Medical Encyclopedia
2626	3.4	http://presidencia.gob.mx/	articles from the website of the President of Mexico
1510	4.3	http://www.ensayistas.org/	essays from the Anthology of Hispanic Essays project
1447	1.8	http://www.multimedios.org/	articles from the Electronic Library of Christianity
1032	1.1	http://www.ficticia.org/	stories from an anthology of edited amateur short stories
916	4.2	http://www.scielo.cl/	scientific journal articles
695	0.6	http://home.cc.umanitoba.ca/f̃ernand4/	short stories
465	7.3	http://www.un.org/	documents from the United Nations archive (reports and parliamentary proceedings)
274	3.3	http://www.cervantesvirtual.com/	classic Hispanic stories and books from the Cervantes Digital Library project

Table 1: Composition of corpus.

words. Therefore the script removed both articles in other languages that were inadvertently collected by the web crawler and portions of articles that consisted of Spanish mixed with another language.

The final corpus after this processing consisted of 35,318,775 words[3]. The composition of the corpus is shown in Table 1. As shown in the table, about 31% of the corpus are news articles, 29% are legal and scientific documents, 22% are essays and encyclopedia articles, 13% are works of literature, and 4.5% are religious documents. The corpus contained 350,592 unique words.

We randomly partitioned the corpus into a training set of about 31 million words and a test corpus of about 4 million words. Partitioning was performed on a per-document basis. Unless otherwise mentioned, all of the experiments in the following sections used this training and test corpus.

3.2 Experimental protocol and evaluation

To test the efficacy of a particular accentuation algorithm, we simply removed all accent marks from the test dataset, ran the algorithm on the stripped corpus, and compared the output of the algorithm to the original (unstripped) corpus. We define accuracy as the fraction of words in the output of an algorithm that exactly match the ground truth.

3.3 Baseline algorithms

To facilitate comparison of experimental results, we use two very simple accent restoration techniques as baselines. The first technique, `no_accents`, does nothing: that is, it simply outputs the stripped words that it receives as input. This method gives an accuracy of 89.222%, or about one error per ten words, on our test dataset. The second technique, `most_likely`, simply chooses the most common accentuation pattern observed in the training data corresponding to a given stripped word. This method achieves an accuracy of 98.818%, or about 11.8 errors per one thousand words.

We note that these extremely simple baseline algorithms achieve surprisingly high results. Since only about 10% of Spanish words have an accent, almost 90% performance can be achieved by simply not restoring any accents at all. Perhaps more surprising is the result that accuracies approaching 99% can be achieved by simply choosing the most common accentuation pattern for a stripped word, regardless of the surrounding context.

[3] This does not include the special "words" representing punctuation marks and numbers; including these tokens, there were 42,618,769 words.

Ambiguity set	Ambiguity type	Best window size (words)
de/dé	preposition/verb	-1
esta/ésta/está	adjective/pronoun/verb	+1
el/él	article/pronoun	+1
llegue/llegué	verb/verb	±15
papa/papá	noun/noun	±7
paso/pasó	verb/verb	±15
publica/pública	verb/noun	±5
que/qué	preposition/interrogative	-2
secretaria/secretaría	noun/noun	±8

Table 2: Optimal Bayesian context window sizes for a few word ambiguity sets.

3.4 Bayesian classifier

The Bayesian classifier requires a parameter that specifies how many words of context surrounding a target word are considered during the disambiguation process. To optimize this parameter, we conducted a series of experiments in which Bayesian classifiers were learned for a range of window sizes. The window sizes we tested were ±1 through ±10, ±12, ±15, and ±30 words. We also tested asymmetric windows: i.e. windows that consider only preceding or subsequent words. We tested asymmetric windows of size -3, -2, -1, $+1$, $+2$, and $+3$. The best accuracy on our corpus was 99.1187%, or about 8.81 errors per thousand words, achieved with a window size of ±2. As the window size increased further, accuracy fell.

It may at first seem surprising that increasing the window size of the Bayesian classifier decreases accuracy. After all, in general we would expect a classifier that uses *more* features to produce *better* results. However, it must be remembered that our Bayesian framework takes an unordered "bag of words" view of the surrounding context. Therefore for large window sizes, words far away from a target word have just as much influence over the classifier's decision as words nearby. In the extreme case of an infinite window size, the probabilities $P(w_j|a_i) = P(a_i)$, and the Bayesian method degenerates into the `most_likely` baseline method.

These results suggest that better automatic accent restoration accuracy can be achieved by adjusting the window size for the Bayesian classifier on a per-word basis. We automatically learned the window sizes in the following way. We trained Bayesian classifiers with many different window sizes, and then applied them to the *training* set and measured the accuracies. Then for each ambiguity set (i.e. stripped word), we chose the window size giving the best performance. Finally, a new single Bayesian classifier was trained that used these optimal window sizes. Table 3.4 shows the optimal window sizes for a few ambiguity sets. The results support our hypotheses in Section 1.1: resolving among words with different parts of speech requires a relatively small window size, whereas resolving among words of the same part of speech requires examining a larger window to choose among words with relatively small semantic differences.

This strategy of choosing the optimal window size on a per-word basis achieved an accuracy of 99.211%, or 7.9 errors per thousand words when applied to the test corpus. This corresponds to a 10% reduction in the error rate compared to the classifier using a fixed window size.

3.5 Hidden Markov Model approach

Words were classified into one of 98 categories using the regular expression pattern matching scheme, as described in Section 2.2. The classification for a given word w was performed as follows. If w exactly matched one of a list of 106 function words, then it was automatically assigned the category of that function word (there were 52 categories of function words). Otherwise, if w's ending matched one of 207 suffixes (grouped into 45 categories), it was assigned the corresponding category. The suffix pattern matching was performed in decreasing order of the suffix length, so that if w matches multiple suffixes, the longest suffix was chosen. If w does not match any of these rules, it is assigned to a default category.

The HMM algorithm achieved 99.0501% accuracy on our test dataset, when trained on our training corpus. This corresponds to an error rate of about 9.50 errors per thousand words. Note that this accuracy is slightly worse than the results obtained using the Bayesian framework. It is also significantly slower,

Method	Test corpus accuracy		Training corpus accuracy		Speed
	(% correct)	(errors/kword)	(% correct)	(errors/kword)	(kwords/second)
Strip	89.2223%	107.78	89.1968%	108.03	163.9
Most likely	98.8180%	11.82	99.0076%	9.92	128.2
Bayesian (s=2)	99.1187%	8.81	99.4393%	5.61	79.2
Bayesian (best s)	99.211%	7.90	99.5267%	4.73	74.8
HMM	99.0501%	9.50	99.2089%	7.91	3.7
HMM + Bayesian	99.2433%	7.57	99.5832	4.17	3.6

Table 3: Comparison of accent restoration performance and running time.

processing about 3.7 thousand words per second compared to 74.8 thousand words per second for the Bayesian framework on an Intel Xeon 3.0GHz system.

3.6 Hybrid approach: combining HMMs and Bayesian classifier

We next tried combining the HMM and Bayesian approaches, as follows. The HMM and Bayesian methods were trained on the training corpus independently, and then each method was run on the *training* corpus. Accuracy statistics were then computed on a per-word (per-ambiguity set) basis. For each stripped word, the most accurate model was chosen and recorded. When applied to a test corpus, the hybrid approach switches between the two models, using the model found to be best during training on a per-word basis.

Table 3 presents the results of the HMM+Bayesian method, including a summary of the other results for comparison purposes. We note that the hybrid method performs better than either the HMM method or the Bayesian method alone, giving a final error rate of about 7.57 errors per thousand words.

3.7 Effect of document type on performance

We next examined algorithm performance by document type. For this experiment, we used the same training corpus as before, but the test corpus was partitioned according to genre. Each partition was then tested individually. Table 4 presents the results, showing the number of errors per 1000 words for each of the individual genres. When interpreting these results, it must be noted that the quantity of training data is *not* uniform across different genres. In other words, we might expect the algorithms to do better on genres that make up a larger share of the training corpus. The first column of 4 shows the percentage of the training data corresponding to each genre of document.

The most difficult document types were the literary categories (modern short stories and classical literature). For these categories, the HMM method performed significantly better than either the Bayesian method or the hybrid “HMM+Bayes” method. We believe this is caused by the fact that literary works tend to use vocabulary that is rare in normal conversation (e.g. descriptive adjectives and verbs, etc.) and complicated grammatical structures (e.g. shifting often between different verb tenses). Because the vocabulary occurs rarely, the Bayesian method probably does not have enough training data to accurately classify many words. On the other hand, the HMM method examines only morphological form, and therefore can still perform well on rare words, as long as the morphological form of the words is common in the language.

The least difficult document type for the HMM+Bayes method were the medical encyclopedia articles, for which an accuracy of 99.75% (2.46 errors per 1000 words) was achieved. Interestingly, for the medical documents the HMM performed significantly worse than the Bayesian framework (9.13 errors per thousand words versus 2.6 errors per thousand words) and even performed worse than the method that simply chooses the most likely accentuation pattern independent of context. We believe these results are explained by the fact that these articles contain many scientific medical terms that are very rare in everyday language. Therefore, for most words, there is only possible accentuation pattern, and so the context-independent maximum likelihood method performs well. On the other hand, the technical terms (including words in English and Latin) are incorrectly categorized by the pattern matching procedure, and so the HMM model performs poorly.

	% of original	Error rate (errors per kword)				
Document type	corpus	Strip	ML	Bayesian	HMM	HMM + Bayes
Modern short stories	1.1%	111.06	21.9	19.07	16.35	18.19
Classical literature	3.3%	110.53	21.01	16.52	13.93	15.45
Encyclopedia articles	2.8%	109.74	13.47	11.99	11.61	11.64
Newspaper articles	8.9%	104.41	11.47	10.01	9.30	9.49
Religious articles	1.8%	104.90	11.29	9.37	8.26	9.04
Scientific articles	9.4%	103.61	9.86	9.43	9.01	9.03
UN documents	7.3%	107.49	3.45	3.29	3.35	3.17
Medical articles	1.7%	121.03	8.77	2.60	9.13	2.46

Table 4: Accent restoration performance by document type.

A final interesting result from this experiment is that for the United Nations documents, the `most_likely` method performs exceptionally well, giving just 3.45 errors per 1000 words. We believe this is because the U.N. documents use a relatively restricted grammatical structure. For example, almost all conjugated verbs in this corpus are in the third-person present or past tense. On the other hand, other genres include other tenses (first-person and second-person tenses frequently occur in quotations, for instance).

3.8 Effect of training corpus size

An important question when using statistical NLP algorithms is how much training data should be collected. On one hand, more training data generally produces better classifiers, but on the other hand, collecting ground truth data can be expensive and labor-intensive. Clearly we want to use the smallest training corpus that still produces acceptable results.

We studied the effect of training corpus size on each of our algorithms. From the training corpus we randomly chose subsets with different numbers of words. We used each subset to train the algorithms, and then tested them on the same 4 million test corpus as before. Figure 1 presents the results of these experiments.

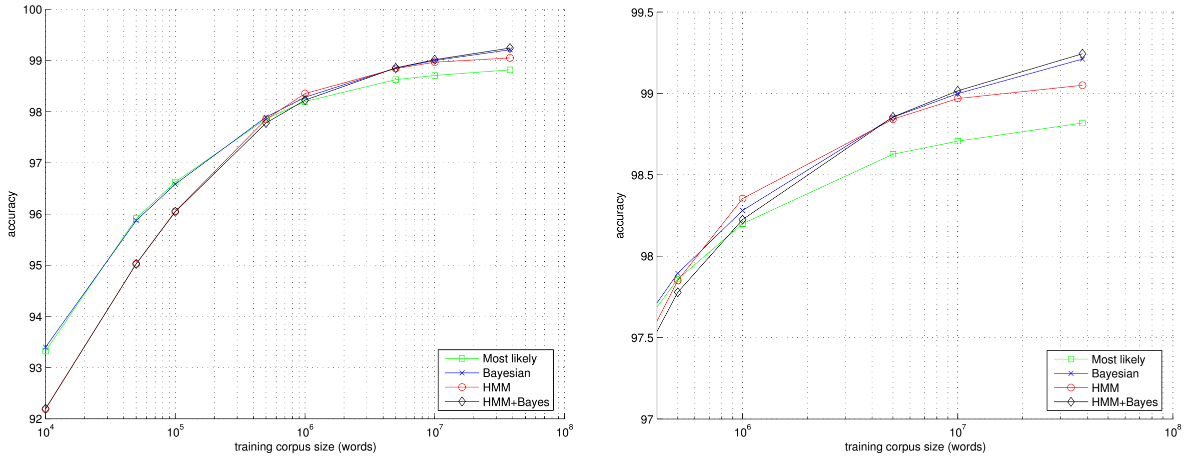
We note that both the HMM and HMM+Bayes methods perform worse than the simple `most_likely` method for small training corpora (less than about 500,000 words). Probably this is explained by the large number of parameters in the HMM and HMM+Bayes models: small amounts of training data are simply insufficient to generate reasonable parameters for these methods. The HMM+Bayes method does not begin to perform better than either individual method until the training corpus reaches about 10 million words. Again, this can be explained by the much larger number of parameters for the HMM+Bayes model.

In future work, we could perform statistical significance tests to be able to draw more firm conclusions from this experiment. However, it appears that the accuracy of the HMM+Bayes method is still increasing even with a training corpus of 35 million words. It would be intersting to try this method on an even larger training corpus of hundreds of millions of words to see how much the accent restoration accuracy can be improved simply by increasing the training corpus size.

3.9 Effect of corpus pollution

It is difficult to collect a Spanish corpus having perfect orthography. Even very well-educated native speakers make accentuation errors; in fact, Yarowski [10] found that even the editors of the Associated Press Spanish newswires made mistakes. Even a writer with perfect orthography may use improper accentuation for effect (i.e. to emulate a character's dialect in a novel) or may directly quote other sources that have imperfect orthography (e.g. music lyrics). Further, it is common for a writer to include words or quotations in languages other than Spanish. All of these factors serve to decrease the quality of the orthography of a corpus, and we refer to these factors collectively as *corpus pollution*.

Corpus pollution of the training set affects the accuracy of a restoration algorithm, since it causes the algorithm to learn sub-optimal training parameters that likely decrease algorithm performance. Corpus



(a) (b)

Figure 1: Effect of training corpus size on each of the algorithms. The plot in (b) is a close-up view of (a).

	Amount of pollution			
Method	0%	1%	5%	10&
Strip	107.78	107.78	107.78	107.78
Most likely	11.82	11.82	11.85	12.58
Bayesian (best w)	7.90	8.24	8.63	9.01
HMM	9.50	9.71	11.192	12.81
HMM + Bayesian	7.57	7.68	8.12	8.89

Table 5: Effect of pollution (incorrectly-accented words) in training corpus on the algorithm performance.

pollution in the test corpus affects the validity of our experiments, since it causes us to compare accentuation results against an imperfect ground truth.

It is straightforward to measure the effect of training corpus pollution on algorithm performance. To do this, we added a controlled amount of pollution to the training corpus, retrained the algorithms, and then re-ran the algorithms on the test dataset. To add pollution, we stripped all accents off 1%, 5%, and 10% of the words in the training corpus. This experiment simulates the effect of imperfect orthography on the algorithm. Table 5 shows the results.

For future work, it would be interesting to pollute the corpus with words from other languages, especially other Romance languages that are similar to Spanish, like French and Portuguese.

It is more difficult to measure the effect of corpus pollution on the test corpus. However, we conducted a small-scale experiment to try to approximate the accuracy of the corpus. A native speaker of Spanish was given a 5,000 word subset of the test corpus and was asked to place accents as needed. On the subset, his accent restoration results agreed with the corpus on 98.185% of the words. When presented with the "correct" accent patterns in the corpus, he concluded that about 0.1% of the words in the corpus were accented incorrectly, and that an additional 0.2% of the words could be correctly accented in more than one way, depending on the exact meaning that the writer had in mind.

These results suggest that a perfect accent restoration algorithm would achieve about 99.7%-99.9% on our test corpus. However, this experiment was so small-scale that the results are probably not statistically applicable to the entire corpus. A better experiment would ask several native speakers to perform the task on a larger subset of the test corpus, and then measure the agreement between speakers. Unfortunately, the amount of human labor involved in such an experiment would be prohibitive for the context of our project. It is clear, however, that achieving 100% accent restoration performance on our imperfect corpus is neither possible nor desirable.

Ambiguity set	Our methods		Yarowsky results	
	Most likely	HMM+Bayes	Most likely	Decision list
anuncio/anunció	52.5%	79.6%	57%	98.4%
registro/registró	78.7%	93.9%	60%	98.4%
marco/marcó	91.0%	94.2%	52%	98.2%
retiro/retiró	74.3%	81.94%	56%	97.5%
regalo/regaló	62.7%	89.8%	56%	90.7%
llegara/llegará	53.2%	74.2%	64%	78.4%
esta/está(/ésta)[4]	57.0%	91.6%	61%	97.1%
mi/mí	84.8%	96.5%	82%	93.7%
secretaria/secretaría	89.6%	91.8%	52%	84.5%

Table 6: Comparison between published results of Yarowsky's decision list algorithm [10] and our HMM+Bayes method for a few ambiguous words. The experiments are not directly comparable, as our task is significantly more difficult than Yarowsky's (see text).

3.10 Comparison to other work

It is difficult to compare our results to other work published in the literature because of differences in corpora and evaluation methodology. These differences are discussed in detail in the literature survey for this project. [2] The only work on Spanish accent restoration that is comparable in scope to our work is Yarowsky's [10]. Unfortunately, they only publish restoration accuracies for a handful of word pairs, instead of accuracy of all words in the corpus, as we have done. However, we can still compare the accuracy of our algorithms on this same handful of words. Table 6 shows the results.

We observe that Yarowsky's method outperforms ours on many of the words, including all those disambiguating between words ending in *o* and *ó*. On the other hand, our approach worked better for disambiguating between different parts of speech (in the case of *mi* and *mí*) and distinguishing the subtle semantic difference between the forms of secretaria.

However, it is important to note that our experiments are not directly comparable to Yarowsky's; in particular, our algorithm is at a disadvantage in this comparison because our experiments are *significantly more challenging* than the ones Yarowsky conducted, in the following ways:

- Yarowsky only stripped accents off the target words in his test corpus, while we stripped accents off *all* words. That is, his accent restoration algorithm considers the *original, properly-accentuated* context around a target word, whereas our algorithm sees only the stripped version of the context. This is gives a significant advantage to his algorithm. When we re-ran the HMM+Bayes method and allowed it to see correct accent patterns of surrounding words, its overall error rate decreased 28% to about 5.5 errors per one thousand words. We believe our experimental setup is better, since in practice, an accent restoration algorithm would not be privy to the correct orthography of surrounding words.

- Yarowsky's algorithm only considers ambiguity pairs, while our algorithm considers ambiguity sets. For example, in the results of Table 6, Yarowsky only considers ambiguity between the pair *esta* and *está*, while we also consider the third possible form, *ésta*. This means Yarowsky's task is significantly easier, since the difference between *ésta* and *esta* is relatively subtle and is confusing even for many native speakers.

- Yarowsky's corpus consists only of newspaper articles, while ours is a diverse collection of many different types of documents. We suspect that our results would improve if training and testing was performed on newspaper articles only.

4 Conclusion

This paper has studied the problem of automatic written accent restoration in Spanish text. We investigated three approaches to solving this problem: a Bayesian approach that uses word co-locations to resolve missing accent ambiguities, an HMM-based approach that restores accents by predicting likely parts-of-speech, and a hybrid approach that applies the better of these two models on a per-word basis. This hybrid approach achieved a restoration accuracy of about 7.6 errors per 1000 words.

We investigated several other factors empirically, like the size of the training corpus and the effect of document type on algorithm performance. We found that the simple context-independent method that simply assigns the most likely accentuation pattern to each word actually performs best when the training corpus size is small (less than 500,000 words). Once the training corpus reaches about 10,000,000 words, the HMM+Bayes method performs significantly better than the other methods. We also found that literary works are the hardest to correctly accentuate, while scientific and legal documents are the easiest.

Many opportunities for future work remain in this area. It would be interesting to build a full part-of-speech tagger using labeled training data to be able to compare our simple approach using pattern matching to a tagger trained in a fully-supervised manner. It would also be interesting to do the tagging with a tri-gram model instead of a bi-gram model. Another possible approach to combining the Bayesian and HMM models would be to have them vote on the correct accent pattern. That is, the likelihood of each accentuation pattern could be computed by combining the likelihoods according to each of the two algorithms, instead of our current approach which consults only one of the two methods. It would also be interesting to create a real-time implementation of the algorithm that could be integrated into a word processing or email software package.

5 Acknowledgments

The author would like to thank Saúl Antonio Blanco Rodriguez for performing the human-corpus agreement experiment described in Section 3.9.

References

[1] I. Bolshakov, A. Gelbukh, and S. Galicia-Haro. A simple method to detect and correct spanish accentuation typos. In *Proc. PACLING-99, Pacific Association for Computational Linguistics*, pages 104–113, 1999.

[2] D. Crandall. Literature survey for cs 674 course project. 2005.

[3] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 133–140, 1992.

[4] M. Davies. Un corpus anotado de 100.000.000 palabras del español histórico y modern. In *Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN 2002)*, 2002.

[5] S. de la Vega and C. Salazar. *Avanzando: Gramática española y lectura.* John Wiley & Sons, Inc., New York, 1997.

[6] S. Galicia-Haro, I. Bolshakov, and F. Gelbukh. A simple spanish part of speech tagger for detection and correction of accentuation error. In *Text, Speech and Dialogue: Second International Workshop, TSD '99*, pages 219–222, 1999.

[7] A. D. M. Simard. Real-time automatic insertion of accents in french text. *Natural Language Engineering*, 7(2):143–165, 2001.

[8] R. Mihalcea and V. Nastase. Letter level learning for language independent diacritics restoration. In *Proceedings of the 6th Conference on Natural Language Learning (CoNNL 20002)*, 2002.

[9] D. Yarowsky. Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. In *Meeting of the Association for Computational Linguistics*, pages 88–95, 1994.

[10] D. Yarowsky. A comparison of corpus-based techniques for restoring accents in spanish and french text. In *Natural Language Processing Using Very Large Corpora*, pages 99–120. Kluwer Academic Publishers, 1999.