Orthographic Case Restoration Using Supervised Learning Without Manual Annotation*

Cheng Niu, Wei Li, Jihong Ding, Rohini K. Srihari
Cymfony Inc.
600 Essjay Road, Williamsville, NY 14221
{cniu, wei, jding, rohini}@cymfony.com

Abstract

One challenge in text processing is the treatment of case insensitive documents such as speech recognition results. The traditional approach is to re-train a language model excluding case-related features. This paper presents an alternative two-step approach whereby a preprocessing module (Step 1) is designed to restore case-sensitive form to feed the core system (Step 2). Step 1 is implemented as a Hidden Markov Model trained on a large raw corpus of case sensitive documents. It is demonstrated that this approach (i) outperforms the feature exclusion approach for Named Entity tagging, (ii) leads to limited degradation for semantic parsing and relationship extraction, (iii) reduces system complexity, and (iv) has wide applicability: the restored text can feed both statistical model and rule-based systems.

1 Introduction

In real-life natural language processing (NLP) applications, a system should be robust enough to handle diverse textual media, degraded to different degrees. One of the challenges from degraded text is the treatment of case insensitive documents such as speech recognition results. In the intelligence domain, the majority of archives consist of documents in all uppercase.

The orthographic case information for written text is an important information source. In particular, the basic information extraction (IE) task Named Entity (NE) tagging relies heavily on the case information for recognizing proper names. Thus when the incoming documents are in normal mixed case, almost all NE systems (e.g. [Bikel *et al* 1999], [Krupka *et al* 1998]) utilize case-related features. When this information source is not available, serious performance degradation will occur, if the system is not re-trained or adapted. In the case of statistical NE taggers, without adaptation, the system simply does not work: the degradation is more than 70% based on our testing. The key issue here is how to minimize the performance degradation by adopting some strategy for system adaptation.

For a system based on language models, a *feature exclusion* approach is used to re-train the models excluding features related to the case information [Miller *et al* 2000, Kubala *et al* 1998, Palmer *et al* 2000]. One argument for this approach is that a case-insensitive NE module can be constructed quickly via re-training. But this is true only if the NE module is entirely based on a statistical model. Some NE systems may not adhere to that model. [Krupka & Hausman 1998] presents an NE tagger based on pattern matching rules; [Srihari, Niu & Li 2000] reports an NE system as a hybrid module consisting of both hand-crafted pattern matching rules and a language model trained by supervised machine learning.

Current research in IE on case insensitive text is restricted to detection of named entities [Robinson *et al* 1999, Palmer *et al* 2000, Kubala *et al* 1998, Chieu and Ng 2002]. When we examine an IE system beyond the shallow processing of NE, the traditional feature exclusion approach may not be feasible. Some sophisticated IE systems involve a full spectrum of linguistic processing in support of relationship extraction and event extraction, as is the case for our NLP/IE system. Each processing module may involve some case information as constraints. It is too costly to maintain two versions of a multi-module system for the purpose of handling two types of incoming documents, with or without case.

Alternatively, a modularized two-step approach is presented in this paper. It consists of a preprocessing module (Step 1) designed to restore case-sensitive form to feed the core system (Step 2). The case restoration module is based on a Hidden Markov Model trained on a large corpus of case sensitive documents, which are drawn from a given domain with no need for human annotation.

To summarize, the two-step approach has a number of advantages over the one-step approach: (i) the training corpus is almost limitless, resulting in a high performance model, with no knowledge bottleneck as faced by many supervised learning scenarios; (ii) the case restoration approach is applicable no matter whether the core system uses a statistical model, a hand-crafted rule system or is a hybrid; (iii) when the core system consists of multiple modules, the case restoration approach relieves the burden of having to re-train or adapt each module; (iv) the two-step approach reduces the system complexity when both case sensitive and case insensitive documents need to be handled,

* This work was partly supported by a grant from the Air Force Research Laboratory's Information Directorate (AFRL/IF), Rome, NY, under contract F30602-02-C-0156. The authors wish to thank Carrie Pine of AFRL for reviewing this work.

the system does not need to keep two models for each module at the same time.

[Gale *et al* 1992] did a preliminary feasibility study for case restoration, using some individual examples. But no substantial research with a large training corpus and full-scale benchmarking has been reported.

The remaining text is structured as follows. Section 2 presents the language model for the case restoration task. Section 3 shows a series of benchmarks for NE tagging, relationship extraction and logical Subject-Verb-Object (SVO) parsing. Section 4 is the conclusion.

2 Implementation of Case Restoration

2.1. Background

The design and implementation of the case restoration module serves as a preprocessing step for a core NLP/IE engine named *InfoXtract*, originally designed to handle normal, case sensitive input.

InfoXtract is a modular, hierarchical NLP/IE system involving multiple modules in a pipeline structure. Figure 1 shows the overall system architecture involving the major modules.

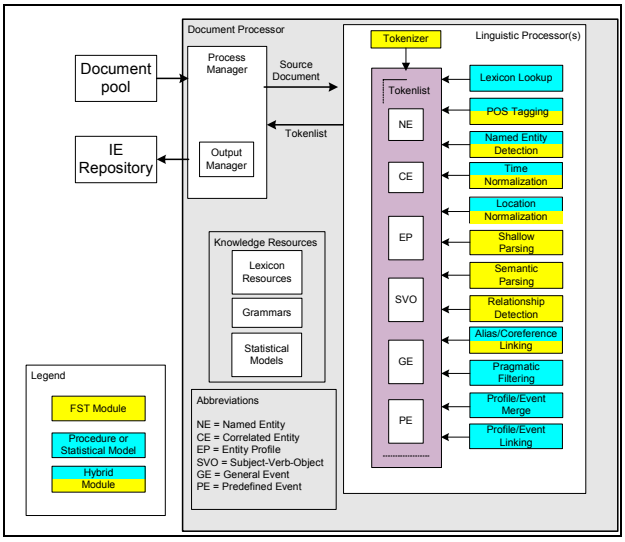

Figure 1: System Architecture of InfoXtract

InfoXtract involves a full spectrum of linguistic processing and relationship/event extraction. This engine, in its current state, involves over 70 levels of processing and 12 major components. Some components are based on hand-crafted pattern matching rules in Finite State Transducer (FST) Modules, some are statistical models or procedures, and others are hybrid (e.g. NE, Co-reference). The major information objects extracted by InfoXtract are NE, Correlated Entity (CE) relationships (e.g. AFFILIATION and POSITION), Entity Profile (EP) that is a collection of extracted entity-centric information, SVO which refers to dependency links between logical subject/object and its verb governor, General Event (GE) on *who did what when and where* and Predefined Event (PE) such as *Management Succession* and *Product Launch*. It is believed that these information objects capture the key content of the processed text. The processing results are stored in IE Repository, a dynamic knowledge warehouse used to support applications.

In order to coordinate with the sophistication of a multi-level NLP/IE system such as InfoXtract, which includes deep parsing and relationship/event extraction capabilities, the restoration approach is not just a recommended option, it is in practice a must. To maintain two versions of such a sophisticated system for the purpose of handling two types of documents, with or without case, is too costly and practically impossible.

Figure2 shows the use of Case Restoration as a plug-in preprocessing module to the core engine. The incoming documents first go through tokenization. In this process, the case information is recorded as features for each token. This token-based case information provides basic evidence for the procedure called Case Detection to decide whether the Case Restoration module needs to be called.

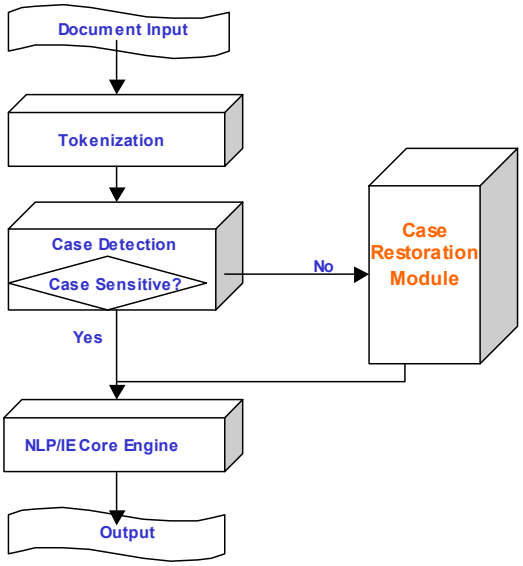

Figure 2: Case Restoration Interfacing NLP/IE Engine

2.2. Implementation

Case restoration is, by nature, a problem at the lexical level; syntactic structures seem to be of no particular help. In [Yarowsky 94], both N-gram context and long distance co-occurrence evidence are used in order to achieve the best performance in *tone restoration*. A similar result was predicted for case restoration. But we observe that the majority of the case restoration phenomena are capturable by local N-gram. We share the observation with [Brill *et al* 1998] that the size of a training corpus is often a more important factor than the complexity of a model for performance enhancement. So a simple bi-gram Hidden Markov Model [Christopher & Hinrich 1999; Bikel *et al* 1999] is selected as the proper choice of language model for this task. Currently, the

system is based on a bi-gram model trained on a normal, case sensitive raw corpus in the chosen domain.

Three orthographic tags are defined in this model: (i) initial uppercase followed by lowercase, (ii) all lowercase, and (iii) all uppercase.[1]

To handle words with low frequency, each word is associated with one of five features: (i) *PunctuationMark* (e.g. &, ?, !…), (ii) *LetterDot* (e.g. A., J.P., U.S.A.,…), (iii) *Number* (e.g. 102,…), (iv) *Letters* (e.g. GOOD, MICROSOFT, IBM, …), or (v) *Other*.

The HMM is formulated as follows. Given a word sequence $W = \langle w_0 f_0 \rangle \cdots \langle w_n f_n \rangle$ (where f_j denotes a single token feature which will be defined below), the goal for the case restoration task is to find the optimal tag sequence $T = t_0 t_1 t_2 \cdots t_n$, which maximizes the conditional probability $\Pr(T \mid W)$ [Bikel 1999]. By Bayesian equality, this is equivalent to maximizing the joint probability $\Pr(W, T)$. This joint probability can be computed by a bi-gram HMM as $\Pr(W,T) = \prod_i \Pr(\langle w_i, f_i \rangle, t_i \mid \langle w_{i-1}, f_{i-1} \rangle, t_{i-1})$. The back-off model is as follows,

$$\Pr(\langle w_i, f_i \rangle, t_i \mid \langle w_{i-1}, f_{i-1} \rangle, t_{i-1})$$
$$= \lambda_1 P_0(\langle w_i, f_i \rangle, t_i \mid \langle w_{i-1}, f_{i-1} \rangle, t_{i-1})$$
$$+ (1 - \lambda_1) \Pr(\langle w_i, f_i \rangle \mid t_i, t_{i-1}) \Pr(t_i \mid w_{i-1}, t_{i-1})$$
$$\Pr(\langle w_i, f_i \rangle \mid t_i, t_{i-1})$$
$$= \lambda_2 P_0(\langle w_i, f_i \rangle \mid t_i, t_{i-1}) + (1 - \lambda_2) \Pr(\langle w_i, f_i \rangle \mid t_i)$$
$$\Pr(t_i \mid w_{i-1}, t_{i-1})$$
$$= \lambda_3 P_0(t_i \mid w_{i-1}, t_{i-1}) + (1 - \lambda_3) \Pr(t_i \mid w_{i-1})$$
$$\Pr(\langle w_i, f_i \rangle \mid t_i)$$
$$= \lambda_4 P_0(\langle w_i, f_i \rangle \mid t_i) + (1 - \lambda_4) \Pr(w_i \mid t_i) P_0(f_i \mid t_i)$$
$$\Pr(t_i \mid w_{i-1}) = \lambda_5 P_0(t_i \mid w_{i-1}) + (1 - \lambda_5) P_0(t_i)$$
$$\Pr(w_i \mid t_i) = \lambda_6 P_0(w_i \mid t_i) + (1 - \lambda_6) \frac{1}{V}$$

Where V denotes the size of the vocabulary, the back-off coefficients λ's are determined using the Witten-Bell smoothing algorithm, and the quantities $P_0(\langle w_i, f_i \rangle, t_i \mid \langle w_{i-1}, f_{i-1} \rangle, t_{i-1})$, $P_0(\langle w_i, f_i \rangle \mid t_i, t_{i-1})$, $P_0(t_i \mid w_{i-1}, t_{i-1})$, $P_0(\langle w_i, f_i \rangle \mid t_i)$, $P_0(f_i \mid t_i)$, $P_0(t_i \mid w_{i-1})$, $P_0(t_i)$, and $P_0(w_i \mid t_i)$ are computed by the maximum likelihood estimation.

A separate HMM is trained for bigrams involving unknown words. The training corpus is separated into two parts, the words occurring in Part I but not in Part II and the words occurring in Part II but not in Part I are all replaced by a special symbol *#Unknown#*. Then an HMM for unknown words is trained on this newly marked corpus. In the stage of tagging, the unknown word model is used in case a word beyond the vocabulary occurs.

[1] The fourth class is mixed case as seen in 'McDonald', 'WordPerfect', etc. As these phenomena are a minority and have little impact on performance and benchmarking, they are excluded from the case restoration training.

3 Benchmark

A series of benchmarks in the context of InfoXtract have been conducted in evaluating the approach presented in this paper. They indicate that this is a simple but very effective method to solve the problem of handling case insensitive input, clearly outperforming the feature exclusion approach.

3.1. Benchmark for Case Restoration

A corpus of 7.6 million words drawn from the general news domain is used in training case restoration. A separate testing corpus of 1.2 million words drawn from the same domain is used for benchmarking. Table 1 shows that the overall F-measure is 98% (P for Precision, R for Recall and F for F-measure).

Table 1: Case Restoration Performance

			P	R	F
Overall			0.96	1	0.98
	Lower Case		0.97	0.99	0.98
	Non-Lower Case		0.93	0.84	0.88
		Initial-Upper Case	0.87	0.84	0.85
		All-Upper Case	0.77	0.6	0.67

The score that is most important for IE is the F-measure of recognizing non-lowercase word. We found the majority of errors involve missing the first word in a sentence due to the lack of a powerful sentence final punctuation detection module in the case restoration stage. But it is found that such 'errors' have almost no negative effect on the following IE tasks.[2]

3.2. Impact of Training Corpus Size

The key for the case restoration approach to work is the availability of a huge training corpus. It is very fortunate that for case restoration, the raw text of normal, case sensitive documents can be used as the training corpus. Such documents are almost limitless, providing an ideal condition for training a high performance system.

In general, the larger the training corpus that is used, the better the resulting model will be [Brill *et al* 1998]. But in practice, we often need to make some trade-off between acceptable performance and available corpus. The corpus size may be limited by a number of factors, including the

[2] This type of 'error' may have a positive effect on NE. The normal English orthographic rule that the first word be capitalized can confuse the NE learning system due to the lack of the usual orthographic distinction between a candidate proper name and a common word.

limitation of the training time and the computer memory, the availability of a case sensitive source that suits the domain, etc. Therefore, it is important to know the growth curve.

Figure 3 shows the impact of the corpus size on the restoration performance. The study shows that the minimum size requirement for training a decent case restoration module is around 2 million words beyond which the performance increase slows down significantly.

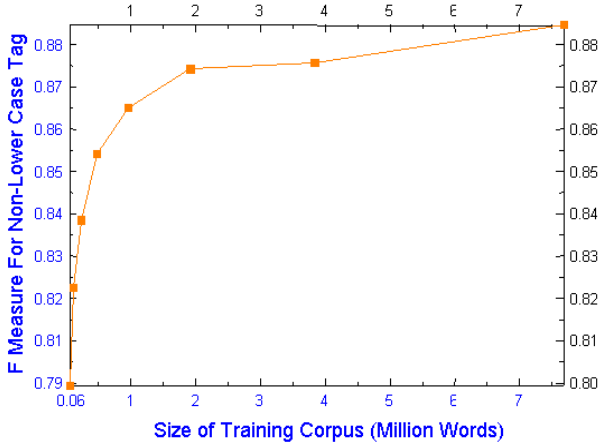

Figure 3: Impact of Training Corpus Size

3.3. Degradation Tests

There is no doubt that the lack of case information from the input text will impact the NLP/IE performance. The goal of the case restoration module is to minimize this impact. A series of degradation tests have been run on three basic IE modules to see how much this impact is and to compare it with the degradation test reported in the literature.

The experiments and the related degradation benchmarks reported below are conducted by using the case restoration module in the context of InfoXtract. We believe that the case restoration module can work equally well for other NLP or IE engines, as this is a preprocessing module, with no dependency on others.

The degradation benchmarking is designed as follows. We start with a testing corpus drawn from normal case sensitive text. We then feed that into InfoXtract for benchmarking. This is normal *baseline* benchmarking for case sensitive text input. After that, we artificially remove the case information by transforming the corpus into a corpus in all uppercase. The case restoration module is then plugged in to restore the case before feeding into InfoXtract. By comparing benchmarking using case restoration with the baseline, we can calculate the level of performance degradation from the baseline in handling case insensitive input for three fundamental capabilities of InfoXtract: NE tagging, CE relationship extraction, and logical SVO parsing (which forms the core of general events). In order to have a direct comparison between the restoration approach and the traditional feature exclusion approach, the performance of our re-trained NE model on case insensitive input is also benchmarked.

3.3.1. NE Tagging Using Case Restoration

An annotated testing corpus of 177,000 words in the general news domain is used for the named entity tagging performance (Table 2), using an automatic scorer following Message Understanding Conference (MUC) NE standards.

Table 2: NE Degradation Benchmarking 1

Type	Baseline			Case Restored		
	P	R	F	P	R	F
TIME	79.3%	83.0%	81.1%	78.4%	82.1%	80.2%
DATE	91.1%	93.2%	92.2%	91.0%	93.1%	92.0%
MONEY	81.7%	93.0%	87.0%	81.6%	92.7%	86.8%
PERCENT	98.8%	96.8%	97.8%	98.8%	96.8%	97.8%
LOCATION	85.7%	87.8%	86.7%	84.5%	87.7%	86.1%
ORG	89.0%	87.7%	88.3%	84.4%	83.7%	84.1%
PERSON	92.3%	93.1%	92.7%	91.2%	91.5%	91.3%
Overall	89.1%	89.7%	**89.4%**	86.8%	87.9%	**87.3%**
Degradation				2.3%	1.8%	**2.1%**

The overall F-measure for NE for the case restored corpus is 2.1% less than the performance of the baseline system that takes the original case sensitive corpus as input.

When the case information is not available, an NE statistical model has to mainly rely on keyword-based features, which call for a much larger annotated training corpus. This is the knowledge bottleneck for all NE systems adopting the Feature Exclusion approach. In order to overcome this bottleneck, Chieu and Ng [2002] proposed to augment the NE training corpus by including machine tagged case sensitive documents. This approach still requires NE re-training, but it improves the model due to its increased training size. It has reported better performance than some previous feature exclusion efforts, with 3%~4% performance degradation. However, due to the noise introduced by the tagging errors, the training corpus can only be augmented by a small fraction (1/8~1/5) with positive effect. So the knowledge bottleneck is still there.

3.3.2. NE Tagging Using Feature Exclusion Re-training

The original statistical NE tagger is based on a Maximum Entropy Markov Model [McCallum *et al* 2000], trained on an annotated corpus of 500,000 words in the general news domain. The NE model is re-trained using the same training corpus, with the case information removed.

Table 3 shows that the degradation for our re-trained NE model is 6.3%, a drop of more than four percentage points when compared with the two-step approach using case restoration. Since this comparison between two approaches is based on the same testing corpus using the same system, the conclusion can be derived that the case restoration approach is clearly better than the traditional feature exclusion approach for NE. This is mainly due to the availability of a huge training corpus from raw text for case restoration (7,600,000 words in our case) and the limited human annotated NE training corpus (500,000 words).

Table 3: NE Degradation Benchmarking 2

Type	Baseline			NE Re-trained		
	P	R	F	P	R	F
TIME	79.3%	83.0%	81.1%	74.5%	74.5%	74.5%
DATE	91.1%	93.2%	92.2%	90.7%	91.4%	91.1%
MONEY	81.7%	93.0%	87.0%	80.5%	92.1%	85.9%
PERCENT	98.8%	96.8%	97.8%	98.8%	96.8%	97.8%
LOCATION	85.7%	87.8%	86.7%	89.6%	84.2%	86.8%
ORG	89.0%	87.7%	88.3%	75.6%	62.8%	68.6%
PERSON	92.3%	93.1%	92.7%	84.2%	87.1%	85.7%
Overall	89.1%	89.7%	**89.4%**	85.8%	80.7%	**83.1%**
Degradation				3.3%	9.0%	**6.3%**

3.3.3. Benchmark on SVO and CE

From an InfoXtract-processed corpus drawn from the news domain, we randomly pick 250 SVO structural links and 60 AFFILIATION and POSITION relationships for manual checking (Table 4).

Surprisingly, there is almost no statistically significant difference in the SVO performance; the degradation due to the case restoration was only 0.07%. This indicates that parsing is less subject to the case factor to a degree that the performance differences between a normal case sensitive input and a case restored input are not obviously detectable.

The degradation for CE is about 6%. Considering there is absolutely no adaptation of the CE module, this degradation is reasonable.

Table 4: SVO/CE Degradation Benchmarking

	SVO			CE		
	Baseline	Case Restored		Baseline	Case Restored	
CORRECT	196	195	Degrad-ation	48	43	Degrad-ation
INCORRECT	13	12		0	1	
SPURIOUS	10	10		2	2	
MISSING	31	33		10	14	
PRECISION	89.50%	89.86%	-0.36%	96.0%	93.5%	2.5%
RECALL	81.67%	81.25%	0.42%	82.8%	74.1%	8.7%
F-MEASURE	**85.41%**	**85.34%**	**0.07%**	**88.9%**	**82.7%**	**6.2%**

4 Conclusion

In order to properly handle case insensitive text, we have presented a case restoration approach by using a statistical model as the preprocessing step for a NLP/IE system. This solution is benchmarked to clearly outperform the traditional feature exclusion approaches for the task of Named Entity tagging:

- for case sensitive input, baseline: 89.4%
- for case insensitive input, using feature exclusion re-training: 83.1%, degradation: 6.3%
- for case insensitive input, using case restoration: 87.3%, degradation: 2.1%

In addition to NE, the SVO parsing and relationship extraction are also tested, with very limited degradation. To our knowledge, this level of NLP/IE has not been benchmarked in case insensitive text before.

Case Restoration presents a rare scenario where supervised learning can be performed with no knowledge bottleneck. A simple statistical bigram technique has been shown to yield very good results.

References

Bikel, D.M., R. Schwartz, R.M. Weischedel. 1999. An Algorithm that Learns What's in a Name. *Machine Learning*, Vol. 1,3, pp. 211-231.

Brill, E., and et al 1998. Beyond N-Grams: Can Linguistic Sophistication Improve Langauge Modeling? *ACL-1998*

Chieu, H.L. and H.T. Ng. 2002. Teaching a Weaker Classifier: Named Entity Recognition on Upper Case Text. *Proceedings of ACL-2002*, Philadelphia.

Christopher, D.M. and S. Hinrich. 1999. *Foundations of Statistical Natural Language Processing*, the MIT press.

Gale, W., K. Church, and D. Yarowsky. 1992. Discrimination Decisions for 100,000-Dimensional Spaces. *Statistical Research Reports*, No. 103, Jan 16, 1992, AT&T Bell Laboratories

Krupka, G.R. and K. Hausman. 1998. IsoQuest Inc.: Description of the NetOwl (TM) Extractor System as Used for MUC-7, *Proceedings of MUC-7*

Kubala, F., R. Schwartz, R. Stone and R. Weischedel. 1998. Named Entity Extraction from Speech. *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop.*

McCallum, A., D. Freitag, and F. Pereira. 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation. *Proceedings of 17th International Conference on Machine Learning*

Miller, D., S. Boisen, R. Schwartz, R. Stone, and R. Weischedel. Named Entity Extraction from Noisy Input: Speech and OCR. *Proceedings of ANLP 2000*, Seattle

Palmer, D., M. Ostendorf, J.D. Burger. 2000. Robust Information Extraction from Automatically Generated Speech Transcriptions. *Speech Communications*, Vol. 32, pp. 95-109.

Robinson, P., E. Brown, J. Burger, N. Chinchor, A. Douthat, L. Ferro, and L. Hirschman. 1999. Overview: Information Extraction from Broadcast News. *Proceedings of The DARPA Broadcast News Workshop Herndon*, Virginia.

Srihari, R., C. Niu, & W. Li. 2000. A Hybrid Approach for Named Entity and Sub-Type Tagging. *Proceedings of ANLP 2000*, Seattle.

Yarowsky, D. 1994. A Comparison Of Corpus-Based Techniques For Restoring Accents In Spanish And French Text. *2nd Workshop on Very Large Corpora*, p319-324.