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 which is subsequently processed by the original system (Step 2). Step 1 is mainly 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 parsing, relationship extraction and case insensitive question answering, (iii) reduces system complexity, and (iv) has wide applicability: the restored text can be used in both statistical model and rule-based systems.

Keywords: Case Insensitive; Case Restoration; Hidden Markov Model; Degraded Text

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 in degraded text is the treatment of case insensitive documents such as speech recognition transcripts. In the intelligence domain, many of the archives consist of documents in all uppercase.

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 case sensitive, almost all NE systems (e.g. *IdentiFinder*,[1] *NetOwl*[2]) 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. The key issue here is how to minimize the performance degradation by adopting some strategy for adapting to the lack of case features.

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.[3 4 5] 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 and Hausman[2] present an NE tagger based on pattern matching rules; Srihari, Niu and Li[6] 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.[4 5 7 8] When we examine an IE system beyond the shallow processing of NE, the traditional feature exclusion approach may not be feasible. Some 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, the system does not need to keep two models for each module at the same time.

Gale *et al* did a preliminary feasibility study for case restoration, using some individual examples.[9] We reported research with a large training corpus and full-scale benchmarking in FLAIRS.[11] In February 2002, our IE engine equipped with the case restoration capability was delivered to a system integrator and in May 2002, the system was deployed to NAIC, a USAF installation for handling case insensitive text. Recently, Lita *et al.* presented similar work as applied to machine translation.[12]

This paper is an extension of our work reported in FLAIRS, including the new research in addressing the issue of irregular mixed-case words using a lexicon acquisition approach (e.g. *McDonald*, *eCommerce, iPod,* etc. see Section 2.2), in resolving case ambiguity using context classification (e.g. *turkey* vs. *Turkey*, see Section 3.3) and in applying case restoration to case insensitive question answering (Section 3.4.4).[13]

The remaining text is structured as follows. Section 2 presents the language model for the case restoration task. Section 3 shows a series of experiments and benchmarks in case restoration and in using case restoration for NE tagging, relationship extraction, parsing and question answering. Section 4 is the conclusion.

2 Implementation of Case Restoration

2.1. *Background*

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.[14] InfoX-tract is a modular, hierarchical NLP/IE system involving multiple modules in a pipeline structure. Fig. 1 shows the overall system architecture involving the major modules.

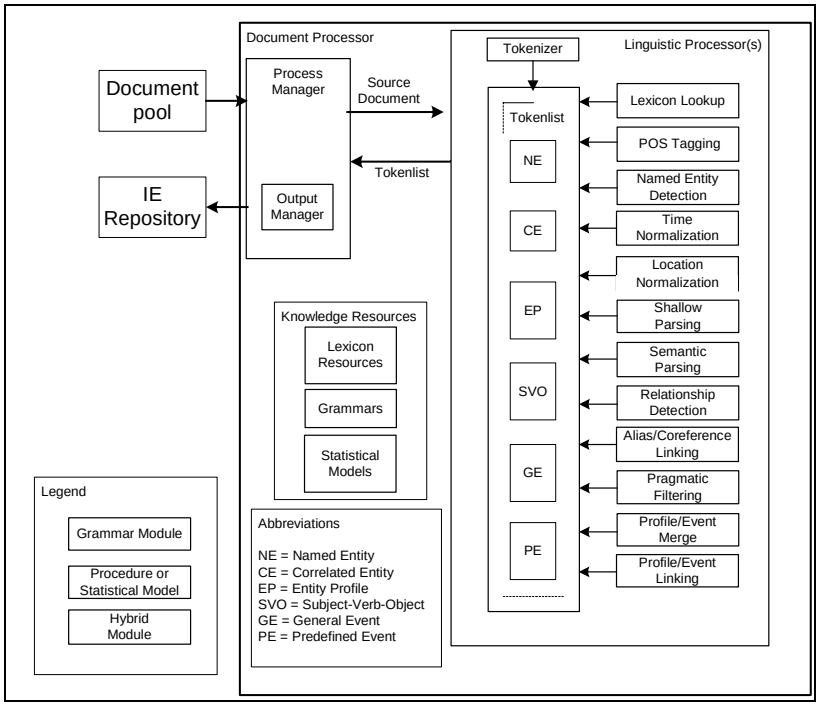

Fig. 1: System Architecture of InfoXtract

InfoXtract involves a full spectrum of linguistic processing and relationship/event extraction. This engine, in its current state, involves over 100 levels of processing and 12 major components. Some components are finite state grammar modules, some are statistical models or procedures, and others are hybrid (e.g. Co-reference). The major information objects extracted by InfoXtract are NEs, Correlated Entity (CE) relationships (e.g. AFFILIATION and POSITION), Entity Profiles (EPs), SVO triples, General Events (GEs) and Predefined Events (PEs).[a] 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 multi-modular system for the purpose of handling two types of documents, with or without case, is too costly and practically impossible.

Fig. 2 shows the use of Case Restoration as a 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

[a] CE relationships include AFFILIATION and POSITION, EP is a collection of extracted entity-centric information, an SVO triple refers to dependency links between logical subject/object and its verb governor, GE represents information on *who did what when and where* and PE includes domain-specific events such as *Management Succession* and *Product Launch*.

basic evidence for the procedure called Case Detection to decide whether the Case Restoration module needs to be called.

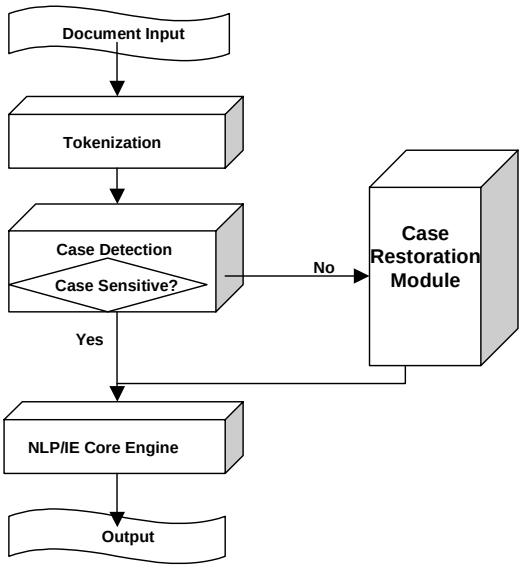

Fig. 2: Case Restoration Interfacing NLP/IE Engine

2.2. *Implementation*

Case restoration is, by nature, a problem at the lexical level; syntactic structures seem to be of no particular help. In the literature, Yarowsky used both N-gram contexts and long distance co-occurrence evidence in order to achieve the best performance in *tone restoration*.[15] A similar result was predicted for case restoration. But we observe that the majority of the case restoration phenomena can be modeled by local N-grams. Based on our experiments, long distance co-occurrence evidence only contributes to the task of case restoration involving some special case-related ambiguous words such as *Turkey* vs. *turkey*. No statistically significant improvement on the overall system performance is observed by bringing in long distance evidence.

We share the observation with Brill *et al.*[16] that the size of a training corpus is often a more important factor than the complexity of a model for performance enhancement. So a bi-gram Hidden Markov Model[1 17] is selected as the proper choice of language model for this task. Our system is based on a bi-gram model trained on a large normal, case sensitive raw corpus in the chosen domain.

Three orthographic tags are defined in this model: (i) all lowercase, and (ii) all uppercase and (iii) mixed-case (involving a mixture of lowercase and uppercase letters). The first two tags have no ambiguity in case restoration. The third tag *mixed-case* can take various orthographic forms. However, the majority of lexical items in mixed-case are in the form of an initial uppercase letter followed by lower case letters, or 'Initial Capitalization'. The exceptional cases include words like *McDonald*, *iMac*, *eBay*, etc. which are recovered using the following lexicon acquisition approach.[b] In a document

recovered using the following lexicon acquisition approach.[b] In a document pool containing 80 millions words, we retrieved a list of 3,144 words which occur in irregular mixed-case in majority of its mentions. When such a word takes different orthographic forms, the most commonly used orthographic form is selected for case restoration. In the tagging stage, we first look up the words in the irregular-case word list. For words not in the irregular word list, the HMM will assign one of the three case tags. Words which are assigned with the mixed-case tag will be restored into the initial capitalization form by default.

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 as defined above), 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 $\Pr(T \mid W)$.[1] 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 using Eq. (1).

$$\Pr(W,T) = \prod_i \Pr(\langle w_i, f_i \rangle, t_i \mid \langle w_{i-1}, f_{i-1} \rangle, t_{i-1}) . \quad (1)$$

The back-off model is shown in Eq. (2) through (7).

$$\begin{aligned} &\Pr(\langle w_i, f_i \rangle, t_i \mid \langle w_{i-1}, f_{i-1} \rangle, t_{i-1}) = \lambda_1 P_0(\langle w_i, f_i \rangle, t_i \mid \langle w_{i-1}, f_{i-1} \rangle, t_{i-1}) \\ &+ (1-\lambda_1)\Pr(\langle w_i, f_i \rangle \mid t_i, t_{i-1})\Pr(t_i \mid w_{i-1}, t_{i-1}) \end{aligned} . \quad (2)$$

$$\Pr(\langle w_i, f_i \rangle \mid t_i, t_{i-1}) = \lambda_2 P_0(\langle w_i, f_i \rangle \mid t_i, t_{i-1}) + (1-\lambda_2)\Pr(\langle w_i, f_i \rangle \mid t_i) . \quad (3)$$

$$\Pr(t_i \mid w_{i-1}, t_{i-1}) = \lambda_3 P_0(t_i \mid w_{i-1}, t_{i-1}) + (1-\lambda_3)\Pr(t_i \mid w_{i-1}) . \quad (4)$$

$$\Pr(\langle w_i, f_i \rangle \mid t_i) = \lambda_4 P_0(\langle w_i, f_i \rangle \mid t_i) + (1-\lambda_4)\Pr(w_i \mid t_i) P_0(f_i \mid t_i) \quad (5)$$

$$\Pr(t_i \mid w_{i-1}) = \lambda_5 P_0(t_i \mid w_{i-1}) + (1-\lambda_5) P_0(t_i) . \quad (6)$$

$$\Pr(w_i \mid t_i) = \lambda_6 P_0(w_i \mid t_i) + (1-\lambda_6)\frac{1}{V} \quad (7)$$

In these equations, V denotes the size of the vocabulary, the back-off coefficients λ's are determined using the Witten-Bell smoothing algorithm, and the following quantities are

[b] The lexicon acquisition approach will generate an exception list of irregular mixed-case words. We observe that there is a certain degree of regularity at morphological level that may be covered more properly by incorporating a morphology analysis program in addition to the lexicon acquisition approach. For example, the use of the prefixes such as *Mc-* (*McDonald, McRobbie,* etc.) can be captured by morphology analysis. This is left for future work.

computed by the maximum likelihood estimation: $P_0(\langle w_i, f_i \rangle, t_i \mid \langle w_{i-1}, f_{i-1} \rangle, t_{i-1})$, $P_0(\langle w_i, f_i \rangle \mid t_i, t_{i-1})$, $P_0(t_i \mid w_{i-1}, t_{i-1})$, $P_0(\langle w_i, f_i \rangle \mid t_i)$, $P_0(f_i \mid t_i)$, $P_0(t_i \mid w_{i-1})$, $P_0(t_i)$, and $P_0(w_i \mid t_i)$.

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 an out-of-vocabulary word occurs.

3 Experiments and Benchmarking

A series of benchmarks have been conducted in evaluating the approach presented in this paper. They indicate that this is a straightforward 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. The irregular mixed-case word list is obtained from a general news corpus of 50 million words. A separate blind 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 97% (P for Precision, R for Recall and F for F-measure).[c]

Table 1. Case Restoration Performance

Case Restored	**P**	**R**	**F**
Overall	0.97	0.97	0.97
All-Lowercase	0.98	0.99	0.98
Non-Lowercase (not distingushing i and ii)	0.92	0.88	0.9
i) All-Uppercase	0.72	0.67	0.7
ii) Mixed-case (not distinguishing iii and iv)	0.9	0.87	0.88
iii) Initial Capitalization	0.9	0.87	0.88
iv) Irregular Mixed-case	0.9	0.59	0.71

Table 2. Case Insensitive Baseline

All Lowercase Output	**P**	**R**	**F**
Overall	0.82	0.82	0.82
All-Lowercase	0.82	1	0.9
Non-Lowercase	N/A	0	N/A

[c] Since the program restores orthographic case for every word, the overall precision and recall are the same value.

The score that is most important for NE is the F-measure (90%) of recognizing 'Non-Lowercase' words. This is because both 'All-Uppercase' (e.g. *IBM*) and 'Mixed-case' (e.g. *Microsoft, eBay*) are strong indicators for proper names while making the distinction between 'All-Uppercase' and 'Mixed-case' is both challenging and often unnecessary. The score of 90% is likely to be under-estimating as we found quite a number of 'Non-Lowercase' errors involve the sentence-initial words due to the lack of a powerful sentence final punctuation detection module in the case restoration stage. It is found that such 'errors' have almost no negative effect on the following IE tasks.[d]

In order to gauge the effect of the case restoration module, we have performed the baseline benchmarking for the case insensitive input, i.e. comparing the (artificially made) all lowercase corpus with the original case sensitive corpus. Table 2 shows the baseline results. The overall performance for the baseline is 82% in F-measure, meaning that in real life case sensitive text, about 82% words are used in all lowercase. The comparison between Table 1 and Table 2 shows that our case restoration module improves the overall performance by 15% in F-measure. More importantly, the case restoration module recognizes non-lowercase words with high performance. This is significant for case insensitive IE, reflected in limited IE degradation (see Section 3.4).

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 most domains, case-sensitive documents are easily available.[e] For example, a news group typically consists of both case sensitive messages and case insensitive/impoverished messages in the same domain: some users are more careful in posting messages in normal case sensitive form and some users are not. We can leverage the case sensitive documents to train a model which can be used to handle case insensitive documents. A simple case detection module can be developed to pick up case sensitive messages to be used as the training corpus for this purpose.

---

[d] In fact, positive effects are observed in some cases. 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.

[e] When available case sensitive documents in the target domain are limited, a mixed corpus that contains both general domain corpus and domain specific corpus can be a feasible approach since any given domain shares a significant portion of common vocabulary with other domains. Another semi-automatic approach in preparing the required training corpus is first to pull a large corpus of case sensitive documents from a source which is closest to the target domain and train a case restoration model. Then we can apply this model to the corpus in the target domain to restore the case. By now the bulk of the case information should have been restored, leaving the remaining problems to be post-edited by human. Obviously, any information analysts or domain specialists who are familiar with the domain can quickly help correct the remaining mistakes. Compared with other annotation tasks such as truthing NE or grammar trees, this task is the least error-prone and has no problem of inter-annotator inconsistency. Simple tools can be developed to help the post-editing as easy as some clicks of mouse to toggle through orthographic case choices in converting to all uppercase, to all lowercase or to initial capitalization. This human post-edited corpus can then be fed back to the training program to learn a high performance, domain specific case restoration system.

In general, the larger the training corpus that is used, the better the resulting model will be.[16] 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 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.

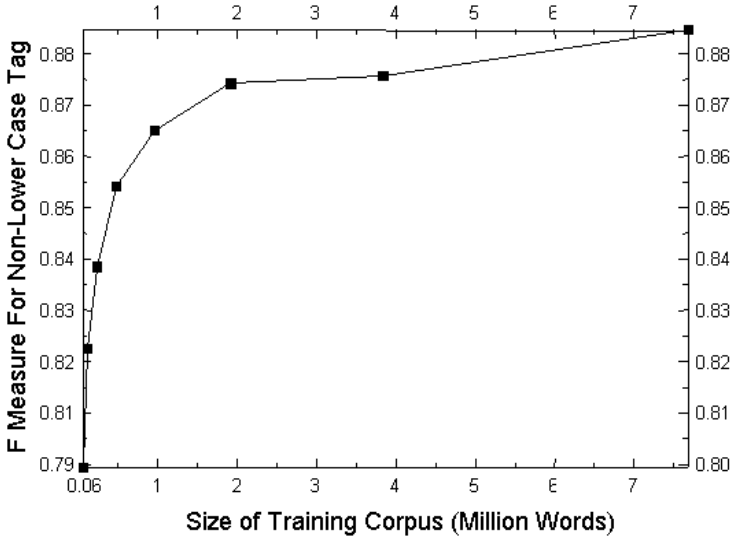

Fig. 3: Impact of Training Corpus Size

Fig. 3 shows the impact of the corpus size on the performance of the HMM (excluding the irregular word list) for recognizing non-lowercase words. The study shows that the minimum size requirement for training a decent case restoration module is around two million words beyond which the performance increase slows down significantly.

3.3. *Contribution of Long Distance Trigger Words to Case Restoration*

There are a small set of words whose orthographic cases refer to different senses of the words. Examples of such words are *China/china*, *Turkey/turkey, White/white*. We call these words case ambiguous words. Case restoration of case ambiguous words is equivalent to course-grained word sense disambiguation (WSD).

Local N-gram context may not be sufficient for WSD; long distance word co-occurrence evidence is often needed.[9] To handle the case ambiguous words, we have implemented a *Naive Bayes*-based context classifier to incorporate the long distance evidence.

First, from WordNet,[10] we retrieve totally 1,181 case ambiguous words. Then we retrieve contexts for each case ambiguous word from a document pool containing 88,000,000 words. Suppose a case ambiguous word *w* is associated with several orthographic cases $\{w_0, w_1, \ldots w_n\}$, then, for each w_i, up to 4,000 snippets containing w_i are

retrieved from the document pool. In this experiment, the range of a snippet is defined as 50 words to the left and 50 words to the right of the case ambiguous word. Sentence-initial case ambiguous words are excluded. Each snippet is regarded as a context of the corresponding word sense, and a Naïve Bayes context classifier[9] is trained based on these snippets. This classifier is used for the task of restoring case for the case ambiguous words.

The case restoration performance of the context classifier is compared with that of our original HMM for the 1,181 case ambiguous words. The overall performance is comparable for these two methods. We only find 248 words where the text classifier clearly out-performs HMM (by at least 5% F-measure) while for the rest of words, HMM either has similar performance to the classifier, or outperforms it. By analyzing the data, we find that in a specific domain, usually one orthographic case for a case ambiguous word is dominant. For example, there are totally 1,125 mentions of *China* vs. only 25 mentions of *china* in the testing corpus. Even all *china* mentions are mis-restored as *China*, the system will still achieve precision around 98%. In such cases, the long distance context is of little help in performance enhancement. We observe that in minority cases, multiple orthographic forms are associated with a case ambiguous word with fairly balanced occurrence frequencies. For example, *Turkey* occurs 347 times while *turkey* occurs 84 times in the testing corpus. The context classifier in this case achieves 93% in disambiguating *Turkey* vs. *turkey* while the HMM achieves 87%. In such cases, using the context classifier is beneficial. Nevertheless, the improvement of the overall system performance is statistically negligible.

3.4. *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 tests in case-insensitive text processing 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 other modules of InfoXtract.

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 benchmarking for normal case sensitive text input as an upper baseline. After that, we artificially remove the case information. The case restoration module is then plugged in to restore the case before feeding into InfoXtract. By comparing benchmarking using case restoration with the normal case sensitive benchmarking, we can calculate the level of performance degradation 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). We have also performed a degradation test on an InfoXtract-supported application in Question Answering (QA).

3.4.1. *NE tagging using case restoration*

We used an annotated testing corpus of 177,000 words in the general news domain, which was originally prepared as a testing corpus in benchmarking our NE tagger on normal case sensitive text. Following Message Understanding Conference (MUC) NE standards, Table 3 shows the NE performance for both normal case sensitive corpus and the case restored corpus using an automatic scorer.

Table 3. NE Degradation Benchmarking

		Normal Case			**Case Restored**		
NE Type		**P**	**R**	**F**	**P**	**R**	**F**
Overall		89.1%	89.7%	**89.4%**	86.8%	87.9%	**87.3%**
	Degradation				2.3%	1.8%	**2.1%**
Name NEs		88.4%	88.5%	**88.5%**	85.4%	86.0%	**85.7%**
	LOCATION	85.7%	87.8%	86.7%	84.5%	87.7%	86.1%
	ORGANIZATION	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%
Non-Name NEs		90.9%	93.5%	**92.2%**	90.8%	93.3%	**92.0%**
	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%

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

3.4.2. *NE tagging using feature exclusion re-training*

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.

The original InfoXtract NE tagger is a statistical model based on a Maximum Entropy Markov Model.[18] This NE tagger is trained on an annotated corpus of 500,000 words in the general news domain. For the purpose of benchmarking, this NE model is re-trained using the same training corpus, with the case information removed.

Table 4 shows that the degradation for our re-trained NE model is 6.3% F-measure, 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 4. NE Degradation for Feature Exclusion Re-training

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%**

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 proposed to augment the NE training corpus by including machine tagged case sensitive documents.[8] 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.

3.4.3. *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 5).

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 5. 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%**

3.4.4. *Benchmark on case insensitive question answering*

Question answering is an application typically supported by NLP and IE. In order to measure the impact of case restoration on an IE-supported QA application, we have experimented with our InfoXtract-QA system[13] and performed the corresponding degradation benchmarking for case insensitive QA.

The corpus from which most QA systems attempt to retrieve answers is usually case sensitive text. However, there are numerous corpora that consist of case insensitive documents, e.g. speech recognition results. In addition, a speech QA application requires that the QA system should plug in a speech recognition interface through which an oral question will be transcribed into case insensitive text. With the case restoration support, the case-restored question and corpus can feed the QA system, which remains unchanged. Experiments show that this approach leads to fairly limited performance degradation from case sensitive QA, mainly due to the limited degradation in the underlying information extraction support. Compared with the baseline case insensitive QA where the case restoration technique is not used, a significant performance enhancement is observed.

The QA experiments were conducted following the TREC-8 QA standards in the category of 250-byte answer strings. In addition to the TREC-8 benchmarking standards Mean Reciprocal Rank (MRR), we also benchmarked precision for the top answer string.

Table 6 shows the results of our first QA experiment where questions remain case sensitive but the TREC corpus is used in three modes for benchmarking. The three modes of the TREC corpus are: (i) case sensitive corpus, (ii) case insensitive corpus, i.e. simply applying the QA system to the same corpus which is artificially made into all lowercase, and (iii) case restored corpus via the case restoration preprocessor. The results on the case restored corpus show very limited performance degradation from the QA on the case sensitive corpus: only 2.8% for MRR and 3.1% for the top answer string. This performance is close to the state-of-the-art case sensitive QA despite the loss of case information in the underlying corpus. Compared with the baseline performance when the

QA is directly applied to the case insensitive corpus, the QA using the case restored corpus shows an enhancement of about 26%. These degradation and enhancement benchmarks demonstrate the power of the case restoration approach in QA applications on case insensitive corpora.

Table 6. QA Using Case Sensitive Questions

Type		Top 1 Precision	MRR
QA on case sensitive corpus (1)		65.7%	73.9%
Baseline QA on case insensitive corpus (2)		36.9%	45.3%
QA on case-restored corpus		62.6%	71.1%
	Degradation from Normal QA (1)	**3.1%**	**2.8%**
	Enhancement to Baseline QA (2)	**25.7%**	**25.8%**

Comparing QA benchmarks with benchmarks for the underlying IE engine shows that the limited QA degradation is in fair proportion with the limited degradation in NE, CE and SVO.

We also conducted an *entirely* case insensitive QA test with case insensitive questions in addition to case insensitive corpus via restoring case for both questions and corpus. This research is meaningful because, when interfacing a speech recognizer to a QA system to accept spoken questions, the case information is not available in the incoming question.[f] We want to know how well the same case restoration technique applies to question processing and gauge the degradation effect on the QA performance and the ways for further enhancement.

Table 7 shows the results of our second QA experiment comparing case insensitive QA with case sensitive QA (on case sensitive corpus, with case sensitive questions). For the case insensitive QA, the performance of the Case Restored QA (on case-restored corpus, with case-restored question) is contrasted with the performance of the Baseline Case Insensitive QA (on case insensitive corpus, with case insensitive questions). Compared with the first experiment where questions remain case sensitive, there is a greater degree of degradation (close to 10%, in contrast to 3%) from the case sensitive QA to the case restored QA while the enhancement to the baseline is also greater (about 32%, in contrast to 26%). Further examination shows that the current case restoration model is still effective but not optimal for question processing, compared with the corpus processing, most probably because the model is not trained on case sensitive question pool. The current case restoration training corpus is drawn from the general news articles which rarely contain questions. If the question case restoration reaches the same performance of the corpus case restoration, the case restored QA system is expected to show further improvement in performance.

[f] In addition to missing the case information, there are other aspects of spoken questions that require treatment, e.g., lack of punctuation marks, spelling mistakes, repetitions. Whether the restoration approach is effective in recovering such information calls for further research, which is beyond the topic in question.

Table 7. Benchmarking-2 for Case Insensitive QA

Type		Top 1 Precision	MRR
Case Sensitive QA (3)		65.7%	73.9%
Baseline Case Insensitive QA (4)		23.2%	32.4%
Case Restored QA		56.1%	64.4%
	Degradation from Normal QA (3)	**9.6%**	**9.5%**
	Enhancement to Baseline QA (4)	**32.9%**	**32.0%**

As question processing results are the starting point and basis for snippet retrieval and feature ranking, an error in question processing seems to lead to greater degradation, as seen in almost 10% drop compared with about 3% drop in the case when only the corpus requires case restoration.

A related explanation for this degradation contrast is as follows. Due to the information redundancy in a large corpus, processing errors in some potential answer strings in the corpus can be compensated for by correctly processed equivalent answer strings. This is due to the fact that the same answer may be expressed in numerous ways in the corpus. Some of the expressions may be less subject to the case effect than others. Question processing errors are fatal in the sense that there is no information redundancy for its compensation. Once it is wrong, it directs the search for answer strings in the wrong direction. Since questions constitute a subset of the natural language phenomena with their own characteristics, case restoration needs to adapt to this subset for optimal performance, e.g. by including more questions in the case restoration training corpus.

In summary, an effective approach to perform case insensitive QA is found with little degradation. This approach uses a high performance case restoration module based on HMM as a preprocessor for the NLP/IE processing of the corpus and the question. There is no need for any changes on the QA system and the underlying IE engine which were originally designed for handling normal, case sensitive corpora. The limited QA degradation is due to the limited IE degradation.

4 Conclusion

In order to properly handle case insensitive text, this paper presents a case restoration approach which uses a statistical model as the preprocessing step for a NLP/IE system. The statistical model is implemented as a bi-gram HMM trained on a large case insensitive corpus, complemented by a lexicon acquisition method for handling words in irregular mixed case (e.g. *McDonald, iPod*). This solution is benchmarked to achieve 97% overall performance in F-measure.

While incorporating a context classifier is found to be beneficial for certain case ambiguous words (e.g. *Turkey* vs. *turkey*), the improvement of the overall system performance is statistically negligible.

The case restoration approach is benchmarked to clearly outperform the traditional feature exclusion re-training approaches for the task of Named Entity tagging. In addition to NE, the SVO parsing, relationship extraction and question answering on case insensitive corpus are also tested using the case restoration support. Compared with processing the case sensitive corpus, there is only limited performance degradation.

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

Acknowledgement

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.

References

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

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

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

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

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

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

[7] P. Robinson, 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.

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

[9] W. Gale, 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

[10] R.Beckwith, C. Fellbaum, D. Gross, and G. A. Miller. 1991. WordNet: A Lexical Database Organized on Psycholinguistic Principles. *Lexicons: Using On-line Resources to build a Lexicon*, Uri Zernik, editor, Lawrence Erlbaum, Hillsdale, NJ.

[11] C. Niu, W. Li, J. Ding and R. Srihari. 2003. Orthographic Case Restoration Using Supervised Learning Without Manual Annotation. *Proceedings of the 16th International FLAIRS Conference 2003*, Florida

[12] L.V. Lita, A. Ittycheriah, S. Toukos and N. Kambhatla. 2003. tRuEcaSing. *Proceedings of ACL-2003*. Sapporo, Japan.

[13] W. Li, R. Srihari, C. Niu, and X. Li. 2003. Question Answering on a Case Insensitive Corpus. *Proceedings of Workshop on Multilingual Summarization and Question Answering - Machine Learning and Beyond* (ACL-2003 Workshop). Sapporo, Japan.

[14] R. Srihari, W. Li, C. Niu and T. Cornell 2003. InfoXtract: A Customizable Intermediate Level Information Extraction Engine. *HLT-NAACL03 Workshop on The Software Engineering and Architecture of Language Technology Systems* (SEALTS), Edmonton, Canada

[15] D. Yarowsky. 1994. A Comparison of Corpus-Based Techniques for Restoring Accents in Spanish and French Text. *2nd Workshop on Very Large Corpora*, p319-324.

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

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

[18] 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*