Analytics for Noisy Unstructured Text Data II

L. Venkata Subramaniam
IBM Research, India Research Lab, India

Shourya Roy
IBM Research, India Research Lab, India

INTRODUCTION

The importance of text mining applications is growing proportionally with the exponential growth of electronic text. Along with the growth of internet many other sources of electronic text have become really popular. With increasing penetration of internet, many forms of communication and interaction such as email, chat, newsgroups, blogs, discussion groups, scraps etc. have become increasingly popular. These generate huge amount of noisy text data everyday. Apart from these the other big contributors in the pool of electronic text documents are call centres and customer relationship management organizations in the form of call logs, call transcriptions, problem tickets, complaint emails etc., electronic text generated by Optical Character Recognition (OCR) process from hand written and printed documents and mobile text such as Short Message Service (SMS). Though the nature of each of these documents is different but there is a common thread between all of these—presence of noise.

An example of information extraction is the extraction of instances of corporate mergers, more formally *MergerBetween(company1,company2,date)*, from an online news sentence such as: *"Yesterday, New-York based Foo Inc. announced their acquisition of Bar Corp."* *Opinion(product1,good)*, from a blog post such as: *"I absolutely liked the texture of SheetK quilts."*

At superficial level, there are two ways for information extraction from noisy text. The first one is cleaning text by removing noise and then applying existing state of the art techniques for information extraction. There in lies the importance of techniques for automatically correcting noisy text. In this chapter, first we will review some work in the area of noisy text correction. The second approach is to devise extraction techniques which are robust with respect to noise. Later in this chapter, we will see how the task of information extraction is affected by noise.

NOISY TEXT CORRECTION

Before moving on to techniques for processing noisy text we will briefly introduce methods for correcting noisy text. One of the most common forms of noise in text is wrong spelling. Kukich provides a comprehensive survey of techniques pertaining to detecting and correcting spelling errors (Kukich, 1992). According to this survey, three types of nonword misspellings are typically found viz. **typographic** such as *teh*, *speel*, **cognitive** such as *recieve*, *conspeeracy* and **phonetic** such as *abiss*, *nacherly*. A distinction must be made between automatically *detecting* such errors and automatically *correcting* those errors. The latter is a much harder problem. Most of the recent work in this area is about correcting spelling mistakes automatically. Golding and Roth (Golding & Roth, 1999) proposed a combination of a variant of *Winnow*, a multiplicative weight-update algorithm and weighted majority voting for context sensitive spelling correction. Mangu and Brill (Mangu & Brill, 1997) have shown that a small set of human understandable rules is more meaningful than a large set of opaque features and weights. Hybrid methods capturing the context using trigrams of the parts-of-speech tags and a feature based method have also been proposed to handle context sensitive spelling correction (Golding & Schabes, 1996). There is a lot of work related to automatic correction of spelling errors (Agirre et. al., 1998), (Zamora et. al., 1983), (Golding, 1995). A complete bibliography of all the work related to spelling error detection and correction can be found in (Beebe, 2005). On a related note, automatic spelling error correction techniques have been applied for other

applications such as semantic role labelling (Sang et. al., 2005).

There is also recent work on correcting the output of SMS text (Aw et. al., 2006) (Choudhury et. al., 2007), OCR errors (Nartker et. al., 2003) and ASR errors (Sarma & Palmer, 2004).

INFORMATION EXTRACTION FROM NOISY TEXT

The goal of Information Extraction (IE) is to automatically extract structured information from the unstructured documents. The extracted structured information has to be contextually and semantically well-defined data from a given domain. A typical application of IE is to scan a set of documents written in natural language and populate a database with the information extracted. The MUC (Message Understanding Conference) conference was one effort at codifying the IE task and expanding it (Chinchor, 1998).

There are two basic approaches to the design of IE systems. One comprises the *knowledge engineering approach* where a domain expert writes a set of rules to extract the sought after information. Typically the process of building the system is iterative whereby a set of rules is written, the system is run and the output examined to see how the system is performing. The domain expert then modifies the rules to overcome any under- or over-generation in the output. The second is the *automatic training approach*. This approach is similar to classification where the texts are appropriately annotated with the information being extracted. For example, if we would like to build a city name extractor, then the training set would include documents with all the city names marked. An IE system would be trained on this annotated corpus to learn the patterns that would help in extracting the necessary entities.

An information extraction system typically consists of natural language processing steps such as morphological processing, lexical processing and syntactic analysis. These include stemming to reduce inflected forms of words to their stem, parts of speech tagging to assign labels such as noun, verb, etc. to each word and parsing to determine the grammatical structure of sentences.

Named Entity Annotation of Web Posts

Extraction of named entities is a key IE task. It seeks to locate and classify atomic elements in the text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. Entity recognition systems either use rule based techniques or statistical models. Typically a parser or a parts of speech tagger identifies elements such as nouns, noun phrases, or pronouns. These elements along with surface forms of the text are used to define templates for extracting the named entities. For example, to tag company names it would be desirable to look at noun phrases that contain the words *company* or *incorporated* in them. These rules can be automatically learnt using a tagged corpus or could be defined manually. Most known approaches do this on clean well formed text. However, named entity annotation of web posts such as online classifieds, product listings etc. is harder because these texts are not grammatical or well written. In such cases reference sets have been used to annotate parts of the posts (Michelson & Knoblock, 2005). The reference set is thought of as a relational set of data with a defined schema and consistent attribute values. Posts are now matched to their nearest records in the reference set. In the biological domain gene name annotation, even though it is performed on well written scientific articles, can be thought of in the context of noise, because many gene names overlap with common English words or biomedical terms. There have been studies on the performance of the gene name annotator when trained on noisy data (Vlachos, 2006).

Information Extraction from OCRed Documents

Documents obtained from OCR may have not only unknown words and compound words, but also incorrect words due to OCR errors. In their work Miller et. al. (Miller et. al., 2000) have measured the effect of OCR noise on IE performance. Many IE methods work directly on the document image to avoid errors resulting from converting to text. They adopt keyword matching by searching for string patterns and then use global document models consisting of keyword models and their logical relationships to achieve robustness in matching (Lu & Tan, 2004). The presence of OCR errors has a detrimental effect on information access