Óscar García Álvarez and Emil Westin

Course: Semantic Analysis

Date: 2015-04-12

Assignment 4: Named Entity Recognition

1. Introduction

This assignment is about the automatic recognition of entities in text. This is an important task for question answering and information retrieval, but it's a field of semantic analysis in itself.

2. Procedures

To solve the task of automatically identifying and classifying entities in a text, we developed a program written in Java. We manually created corpora by retrieving names of persons, locations and companies from various sources from the web, such as Wikipedia and other databases. These collections of texts serve as identifying entities in a text.

In our program, we have chosen to detect four different types of entities: names and surnames for persons, locations and companies.

In order to make the program fast enough, we didn't try to recognize words in lower case letters. To recognize every word in a text can take a lot of time. We are not trying to make an accurate program, because we decided it's not worth it.

We didn't try to recognize initials as part of a name, but we have included various famous initials for cities (LA, NYC and DC). This is because it was difficult to introduce all possible ways to write the name and the surname, especially if we think of Spanish tradition, where they use two surnames.

We decided not to include institutions for the same reason. The structure is really complicated and can include words such as "for", "of", "the" etc., which are difficult to identify inside of the name of an entity, because they are not marked with capital letters which are useful signs for the system.

The difficulties of entity recognition are plenty:

- *Recognition of initials:* the different ways of punctuations is the smallest of the problem. The real question is to get the meaning from the context. It can be quite easy to distinguish if it's referred to a person if it is close to a name or surname, but it's almost impossible to distinguish from a list if it's a person, a place or a company (universities used to be called by their initials more than its full name, for example).

- *Ambiguity of names:* for example, a name of a person can also be the name of a city, like "Santa Barbara", or more often the surname. The same situation can occur with the names of institutions or even companies.
- *The recognition of the length of the entity:* it can be confusing for the system to decide the length of the entity, for example in the case of "York" and "New York". In this case, it is clear that the system has to take the longest matching (greedy inference), but it can lead to errors. For example, if we have a sentence "Smith, NY", the greedy inference might assume that NY are the initials of the name, but in this case they are clearly the initials of a place.
- *The relation between the entities:* the difference between entities sometimes is just grammatical, for example "York" is a city, "Yorktown University" is an institution and "professor in Yorktown University" is a person.
- *Long distance antecedents:* the entities can also be referred to in different ways in a text. For example, "Santa Barbara University" could be referred to with a possessive "Barbara's". The system will probably ignore the possessive and identify it as a name instead of an institution.

First of all, we didn't have an annotated corpus to work with, so we decided to work with dictionaries (word lists). Because we used hashmaps, the system couldn't recognize entities with possessives, but we erased the punctuations at the end of words in order to identify them except initials.

Secondly, we decided to label all the words which have at least one uppercase letter. Our system distinguished between initials, capitalized, all caps, mixed case, ends in digit and contains hyphen. We didn't use this distinction, but it could be useful in order to try to identify initials by context. These are the words we would work with.

After the labeling, the system tried to identify the word as a person, a place or a company (proper nouns). Even if the system used the greedy inference it doesn't join names and surnames in the same entity. We didn't divide our lists according to the number of words which could be a clear improvement.

2.1 Results

The program is efficient and recognizes the words in the different lists. Because of the big memories of nowadays computers, the size of the list is not a big deal, but on the other side the bigger the list (more accuracy) the more time it takes to analyze (less efficient). To analyze the first 10 texts from the corpus we are provided, it takes about 1,83 seconds on my computer. If we try to analyze the whole corpus, it takes about 35 minutes. This is due to the size of the corpus and the fact that there are a lot of capitalized words directly after a full stop, which multiplied the words to analyze but they are impossible to avoid working with.

We didn't consider calculating the accuracy and the precision because we knew that we couldn't identify the institutions, nor join names and surnames together, which means that we don't unify entities. So it's not fair to talk about accuracy or precision.

3. Conclusions

The first conclusion can be that dictionaries are not efficient in terms of time, but still offer an easy way to recognize some entities. On the other side, we have to point out that even if you use dictionaries you have to cross them, in order to avoid the ambiguities. Then a way to evaluate which is the correct label that cases must be implemented. Obviously, the decision of the label cannot be made by a list criteria, another approach is required.

Another question is about efficiency. We know that word lists are not fast enough but we wonder if to work with corpus machine learning techniques are more efficient or less.

Another question which is important to analyze is about the structure of the entities. We have said above that the name of a person can be presented in different ways. If that's true for the name of the persons, it seems very difficult to create a list of all possible structures for institutions. Jurafsky and Martin seem to suggest that syntactic analysis improve the results but it's hard to believe that it can be made in a more efficient way.

The problem with initials is so difficult that we have just avoided it. We should say the same about long distance antecedents. We can't find any good solution for that anyway.

The recognition of the length of the entity is also a problem in the classification. We found out that the greedy inference works fairly well for the longest matching sequence, but that it sometimes can make errors. We think that the percentage of the error that the inference introduces in the system is worthy in terms of the problem it solves.