BUAP_1: A Naïve Approach to the Entity Linking Task

David Pinto, Mireya Tovar, Darnes Vilariño, Beatriz Beltrán, Josefa Somodevilla
Faculty of Computer Science, B. Autonomous University of Puebla
{dpinto, mtovar, darnes, bbeltran, mariasg}@cs.buap.mx

Abstract. In these notes we are reporting the obtained results by applying the Naïve Bayes classifier to the Entity Linking task of the Knowledge Base Population track at the Text Analysis Conference. Three different runs were submitted to the challenge, each with different ways of approaching the application of the above mentioned classifier. The obtained results were very low, and recent analyses showed that this issue was derived from errors at the pre-processing stage.

i. Introduction

The aim of the Knowledge Base Population (KBP) track at the Text Analysis Conference (TAC) was to encourage the evaluation of computational systems which automatically discover information about named entities with the ultimate goal of integrate this information into a knowledge database. For this purpose, an initial Knowledge Base (KB) with multiple entities derived from Wikipedia was provided, together with a set of queries which must be resolved according to the defined tasks. In general, the KBP track was conformed of two tasks which are described as follows:

The Entity Linkage (EL) task: It consists on determining, for a given query, whether or not some knowledge base entity is being referred to. In case this entity exists, then the link is done by returning as result the pair *QueryID-EntityID*, otherwise a pair with the NIL string attached (*QueryID*-“*NIL*”) is given. Each query is provided with three parameters: QueryID, name-string and reference-document. The purpose is to provide further information about this specific query (name-string) in order to disambiguate it, if needed. A major description of data is given in Section 2.

The Slot Filling (SF) task: It consists on learning a pre-defined set of relationships and attributes for target entities based on a set of documents provided on the basis of a set of queries which contain a *name-string*, *docid*, *entity-type*, *node-id* and a *list of slots* to ignore. Slot Filling involves mining information about entities from text and may be viewed as more traditional Information Extraction, or alternatively, as a Question Answering (QA) task, where the questions are static but the targets change.

Even if these two tasks were presented at TAC, we only participated with the former, i.e., the entity linking task.

The rest of this paper is structured as follows. Section 2 presents a description of data used in this approach. The naïve approach used in this paper is described in Section 3. In Section 4 we show the obtained results. Finally, in Section 5, a discussion of the experiment is given.

ii. Datasets

In order to evaluate the entity linking task, two major datasets were provided. Firstly, the Knowledge Base (KB) which is going to be used as target for queries related with some entity of this KB. Secondly, the queries with corresponding reference documents were given. An overview of these datasets is presented as follows.

The knowledge base contains a set of entities, each with a canonical name and title for the Wikipedia page, an entity type, an automatically parsed version of the data from the infobox in the entity's Wikipedia article, and a stripped version of the text of the Wiki article. The Wikipedia infoboxes and entries are taken from an October 2008 snapshot of Wikipedia. A subset of these entities will have mentions in the Entity Linking target list. Representation in the TAC 2009 KBP Evaluation Source Data corpus was not a criterion for inclusion in the Reference KB.

A set of 818,741 entities were provided in the KB. Each entity in the knowledge base is assigned one of four types: PER, ORG, GPE and UKN. It is remarkable the difference with the number of items available for UKN with respect to the other categories (see Table 1). The high unbalanced situation of this training corpus encourages using ad-hoc techniques in order to avoid an over-representation of the UKN category.

Table 1. The Knowledge Base for the KBP track at TAC 2009.

Entity type	Entity description	Number of entities
PER	person	114,523
ORG	organization	55,813
GPE	geo-political entity	116,498
UKN	unknown	531,907

Figure 1 shows a sample of the KB xml file. Each entity is given with four attributes: **wiki_title**, **type** (see Table 1), **id** and **name**. The Wikipedia infobox follows marked by the **facts** tag. Finally, the text associated to the entity is given embraced by the **wiki_text** tag.

```
<entity wiki_title="Mike_Quigley_(footballer)" type="PER" id="E0000001"
name="Mike Quigley (footballer)">
<facts class="Infobox Football biography">
<fact name="playername">Mike Quigley</fact>
<fact name="fullname">Michael Anthony Joseph Quigley</fact>
<fact name="dateofbirth">October 2, 1970 (1970-10-02) (ageÂ 38)</fact>
<fact name="cityofbirth"><link entity_id="E0467057">Manchester</link></fact>
<fact name="countryofbirth"><link entity_id="E0145816">England</link></fact>
<fact name="position"><link>Midfielder</link></fact>
:
</facts>
<wiki_text><![CDATA[Mike Quigley (footballer)
Mike Quigley (born 2 October 1970) is an English football midfielder.
]]></wiki_text>
</entity>
```

Figure 1. Content of the xml file with entities of the KB provided.

With respect to the set of queries provided for the evaluation of the EL task, in Figure 2 it is presented a sample of the Entity Linking List, i.e., the set of queries to be evaluated. As can be seen, the xml tags **query**, **name** and **docid** are given. Thus, these tags correspond to the query id, the query string (name-string) and the document to be used as reference, respectively.

```
<?xml version='1.0' encoding='utf8'?>
<kbpentlink>
 <query id="EL1">
  <name>Abbas Moussawi</name>
  <docid>LTW_ENG_19960311.0047.LDC2007T07</docid>
 </query>
 <query id="EL2">
  <name>Abbas Moussawi</name>
  <docid>NYT_ENG_20000711.0026.LDC2007T07</docid>
 </query>
:
```

Figure 2. Content of the xml file with queries provided.

iii. The Naïve Approach

In the experiments carried out, we have selected the Naïve Bayes classifier in order to train data from a knowledge base and, thereafter, to solve the problem of entity linking through the classification of queries.

The Naïve Bayes is a very simple probabilistic classifier [1] which is based on the Bayes' theorem and complemented with strong (naïve) independence assumptions. In terms of document classification, it is assumed that each term (word or feature F_i) occurrence does not depend in any way on the term occurrence of other terms, i.e., if we have a category C, then $p(F_i|C,F_{i-1}) = p(F_i|C)$.

Formally, given a class (category) C and a set of features (F_1, …, F_n), the conditional distribution over the class variable C can be expressed like this:

$$p(C|F_1,\dots,F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C)$$

where Z is a scaling factor dependent only on F_1, …, F_n, i.e., a constant if the values of the feature variables are known.

The naïve assumption provides a manageable model for classification, and we may estimate the probability of each term by simply dividing each frequency between the total frequency of terms in the category. In Figure 3 we show the general process of classification when using the Naïve Bayes classifier with the KBP data provided. This process involves two stages: training and test. In the former case, a training data is supplied to the classifier which constructs a model of classification which is further used with the test data in order to find the target category for each query.

Following we describe the two approaches implemented for solving the entity linking task by means of classifying queries to categories tagged by the KB entity ids.

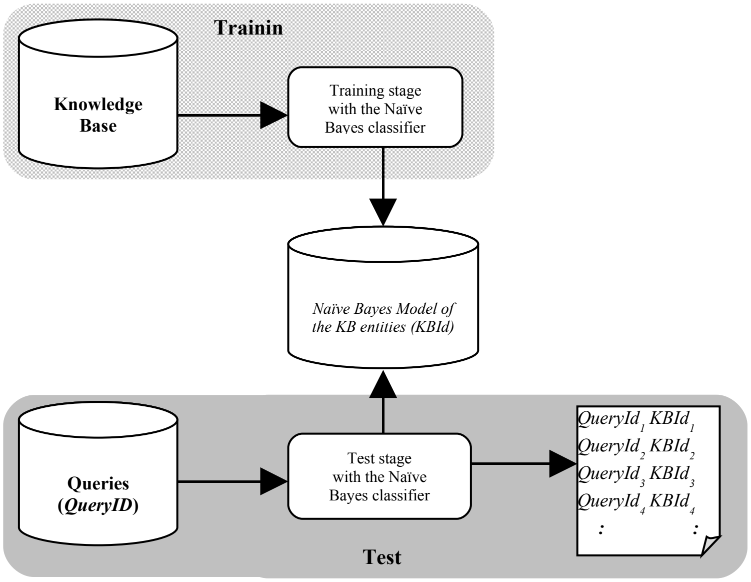
Trainin
Knowledge Base
Training stage with the Naïve Bayes classifier
Naïve Bayes Model of the KB entities (KBId)
Queries (QueryID)
Test stage with the Naïve Bayes classifier
QueryId1 KBId1
QueryId2 KBId2
QueryId3 KBId3
QueryId4 KBId4
: :
Test

Figure 3. Overview of the classification process with the Naïve Bayes classifier.

Approach number 1 (Two phase training**):** In this approach we are interested on discarding first those KB entities that do not agree with the named entity of the query given. Thus, we first train the classifier using the NE type as category. Once we know with certain probability that the query is some specific type of entity (PER, ORG, GPE and UKN). We then proceed with a second phase, where we train an additional classifier with only those KB entities which match the entity type. We use the KB entity **id** as category when training with the last classifier. The best performance is obtained classifying named entities, the best averages results the overall performance will be. The process is represented in Figure 4.

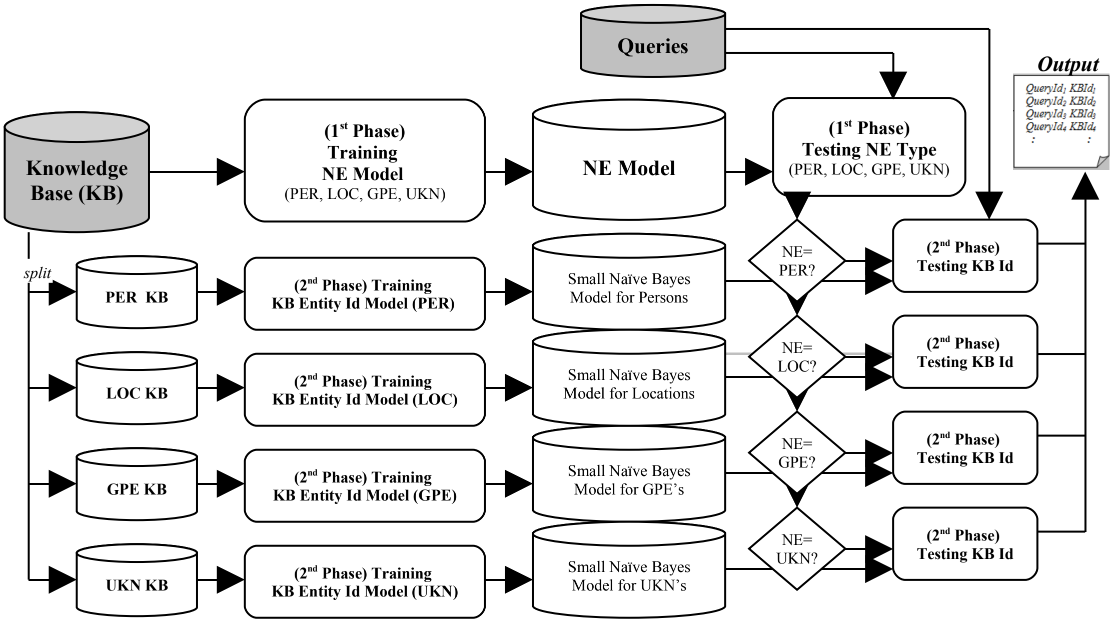
Queries
Output
QueryId1 KBId1
QueryId2 KBId2
QueryId3 KBId3
QueryId4 KBId4
: :
Knowledge Base (KB)
(1st Phase) Training NE Model (PER, LOC, GPE, UKN)
NE Model
(1st Phase) Testing NE Type (PER, LOC, GPE, UKN)
split
PER KB
(2nd Phase) Training KB Entity Id Model (PER)
Small Naïve Bayes Model for Persons
NE= PER?
(2nd Phase) Testing KB Id
LOC KB
(2nd Phase) Training KB Entity Id Model (LOC)
Small Naïve Bayes Model for Locations
NE= LOC?
(2nd Phase) Testing KB Id
GPE KB
(2nd Phase) Training KB Entity Id Model (GPE)
Small Naïve Bayes Model for GPE's
NE= GPE?
(2nd Phase) Testing KB Id
UKN KB
(2nd Phase) Training KB Entity Id Model (UKN)
Small Naïve Bayes Model for UKN's
NE= UKN?
(2nd Phase) Testing KB Id

Figure 4. Two phase classification with the Naïve Bayes classifier.

Approach number 2 (Full training**):** Each KB entity **id** is used as category when training with the Naïve Bayes classifier. In other words, we are expecting a direct linking between the query and the target KB entity. The process is represented in Figure 3. The hypothesis is that this approach would performance worse than the approach number 1.

iv. Experimental results

In the experiments we have carried out, we considered to use three different representations for the KB data: *infobox*, *wiki_text*, and *complete* (*infobox*+*wiki_text*), in order to validate which one obtains the best performance by using the same classifier. Since only three different runs were allowed at the competition, we submitted the following ones:

Run1: Infobox representation with approach number 1.
Run2: Complete representation (Infobox+text) with approach number 1.
Run3: Infobox representation with approach number 2.

Table 2 and 3 show the obtained Micro and Macro averages, respectively. The values presented where undoubtedly the lowest ones at the official table of results of the entity linking task at KBP. There is one mainly reason of the poor performance obtained by the classifier. An error in the pre-processing stage leaded to consider only one of the 88 xml files provided for the knowledge base. In other words, we trained the classifier with only 6,422 entities from the 818,741 which are the total that should be considered. This error forced the classifier to categorize queries to one of these 6,422 entities, ignoring the 812,319 remaining ones. The correct averages will be presented in a poster at the TAC 2009 Workshop. It is only with these correct values as we may analyse the real performance of the presented approaches.

Table 2. Micro-averages obtained by the three approaches in the EL task.

	3904 queries	**1675 non-NIL**	**2229 NIL**
Run1	0.0031	0.0000	0.0054
Run2	0.0020	0.0006	0.0031
Run3	0.0085	0.0006	0.0144

Table 3. Macro-averages obtained by the three approaches in the EL task.

	560 entities	**182 non-NIL**	**378 NIL**
Run1	0.0037	0.0000	0.0055
Run2	0.0057	0.0055	0.0059
Run3	0.0112	0.0055	0.0140

v. Discussion

We have tackled the problem of Entity Linking by means of Entity classification. Two different approaches were implemented. Both based on the Naïve Bayes classifier. The reported results are useless since there were an error on the pre-processing stage. There still however the discussion about the performance of the presented approaches when the data is processed correctly.

Moreover, we are interested on analyzing the behaviour of the implemented classifiers when using: a) only infoboxes information, b) using only wiki_text information and c) using the complete knowledge base information (infobox+wiki_text).

vi. References

[1] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification (2nd ed.)*, John Wiley and Sons, 2001.