Improving Toponym Disambiguation by Iteratively Enhancing Certainty of Extraction

Mena B. Habib and Maurice van Keulen
Faculty of EEMCS, University of Twente, Enschede, The Netherlands

Keywords: Named Entity Extraction, Named Entity Disambiguation, Uncertain Annotations.

Abstract: Named entity extraction (NEE) and disambiguation (NED) have received much attention in recent years. Typical fields addressing these topics are information retrieval, natural language processing, and semantic web. This paper addresses two problems with toponym extraction and disambiguation (as a representative example of named entities). First, almost no existing works examine the extraction and disambiguation interdependency. Second, existing disambiguation techniques mostly take as input extracted named entities without considering the uncertainty and imperfection of the extraction process.
It is the aim of this paper to investigate both avenues and to show that explicit handling of the uncertainty of annotation has much potential for making both extraction and disambiguation more robust. We conducted experiments with a set of holiday home descriptions with the aim to extract and disambiguate toponyms. We show that the extraction confidence probabilities are useful in enhancing the effectiveness of disambiguation. Reciprocally, retraining the extraction models with information automatically derived from the disambiguation results, improves the extraction models. This mutual reinforcement is shown to even have an effect after several automatic iterations.

1 INTRODUCTION

Named entities are atomic elements in text belonging to predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. Named entity extraction (a.k.a. named entity recognition) is a subtask of information extraction that seeks to locate and classify those elements in text. This process has become a basic step of many systems like Information Retrieval (*IR*), Question Answering (*QA*), and systems combining these, such as (Habib, 2011).

One major type of named entities is the toponym. In natural language, *toponyms* are names used to refer to locations without having to mention the actual geographic coordinates. The process of *toponym extraction* (a.k.a. toponym recognition) aims to identify location names in natural text. The extraction techniques fall into two categories: rule-based or based on supervised-learning.

Toponym disambiguation (a.k.a. toponym resolution) is the task of determining which real location is referred to by a certain instance of a name. Toponyms, as with named entities in general, are highly ambiguous. For example, according to GeoNames[1], the toponym "Paris" refers to more than sixty different geographic places around the world besides the capital of France. Figure 1 shows the top ten of the most ambiguous geographic names. It also shows the long tail distribution of toponym ambiguity and the percentage of geographic names with multiple references.

Another source of ambiguousness is that some toponyms are common English words. Table 1 shows a sample of English-words-like toponyms along with the number of references they have in the GeoNames gazetteer.

Table 1: A Sample of English-words-like toponyms.

And	2	The	3
General	3	All	3
In	11	You	11
A	16	As	84

A general principle in our work is our conviction that Named entity extraction (NEE) and disambiguation (NED) are highly dependent. In previous work (Habib and van Keulen, 2011), we studied not only

[1] www.geonames.org

B. Habib M. and van Keulen M..
Improving Toponym Disambiguation by Iteratively Enhancing Certainty of Extraction.
DOI: 10.5220/0004174903990410
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (SSTM-2012), pages 399-410
ISBN: 978-989-8565-29-7

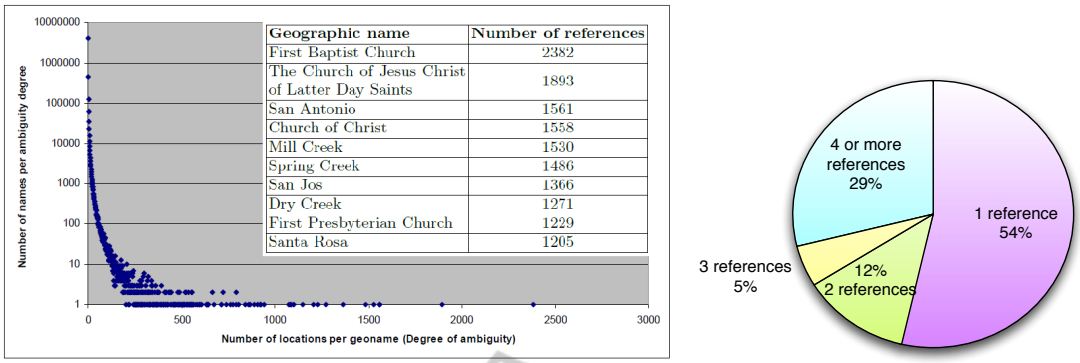


Figure 1: Toponym ambiguity in GeoNames: top-10, long tail, and reference frequency distribution.

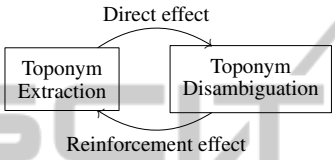


Figure 2: The reinforcement effect between the toponym extraction and disambiguation processes.

the positive and negative effect of the extraction process on the disambiguation process, but also the potential of using the result of disambiguation to improve extraction. We called this potential for mutual improvement, the *reinforcement effect* (see Figure 2).

To examine the reinforcement effect, we conducted experiments on a collection of holiday home descriptions from the EuroCottage[2] portal. These descriptions contain general information about the holiday home including its location and its neighborhood (See Figure 4 for an example). As a representative example of toponym extraction and disambiguation, we focused on the task of extracting toponyms from the description and using them to infer the country where the holiday property is located.

In general, we concluded that many of the observed problems are caused by an improper treatment of the inherent ambiguities. Natural language has the innate property that it is multiply interpretable. Therefore, none of the processes in information extraction should be 'all-or-nothing'. In other words, all steps, including entity recognition, should produce *possible* alternatives with associated likelihoods and dependencies.

In this paper, we focus on this principle. We turned to statistical approaches for toponym extraction. The advantage of statistical techniques for extraction is that they provide alternatives for annotations along with confidence probabilities (confidence for short). Instead of discarding these, as is commonly done by selecting the top-most likely candidate, we use them to enrich the knowledge for disambiguation. The probabilities proved to be useful in enhancing the disambiguation process. We believe that there is much potential in making the inherent uncertainty in information extraction explicit in this way. For example, phrases like "Lake Como" and "Como" can be both extracted with different confidence. This restricts the negative effect of differences in naming conventions of the gazetteer on the disambiguation process.

Second, extraction models are inherently imperfect and generate imprecise confidence. We were able to use the disambiguation result to enhance the confidence of true toponyms and reduce the confidence of false positives. This enhancement of extraction improves as a consequence the disambiguation (the aforementioned reinforcement effect). This process can be repeated iteratively, without any human interference, as long as there is improvement in the extraction and disambiguation.

The rest of the paper is organized as follows. Section 2 presents related work on NEE and NED. Section 3 presents a problem analysis and our general approach to iterative improvement of toponym extraction and disambiguation based on uncertain annotations. The adaptations we made to toponym extraction and disambiguation techniques are described in Section 4. In Section 5, we describe the experimental setup, present its results, and discuss some observations and their consequences. Finally, conclusions and future work are presented in Section 6.

2 RELATED WORK

NEE and NED are two areas of research that are well-covered in literature. Many approaches were developed for each. NEE research focuses on improving

[2]http://www.eurocottage.com

the quality of recognizing entity names in unstructured natural text. NED research focuses on improving the effectiveness of determining the actual entities these names refer to. As mentioned earlier, we focus on toponyms as a subcategory of named entities. Is this section, we briefly survey a few major approaches for toponym extraction and disambiguation.

2.1 Named Entity Extraction

NEE is a subtask of Information Extraction (IE) that aims to annotate phrases in text with its entity type such as names (e.g., person, organization or location name), or numeric expressions (e.g., time, date, money or percentage). The term 'named entity recognition (extraction)' was first mentioned in 1996 at the Sixth Message Understanding Conference (MUC-6) (Grishman and Sundheim, 1996), however the field started much earlier. The vast majority of proposed approaches for NEE fall in two categories: handmade rule-based systems and supervised learning-based systems.

One of the earliest rule-based system is FASTUS (Hobbs et al., 1993). It is a nondeterministic finite state automaton text understanding system used for IE. In the first stage of its processing, names and other fixed form expressions are recognized by employing specialized microgrammars for short, multiword fixed phrases and proper names. Another approach for NEE is matching against pre-specified gazetteers such as done in LaSIE (Gaizauskas et al., 1995; Humphreys et al., 1998). It looks for single and multi-word matches in multiple domain-specific full name (locations, organizations, etc.) and keyword lists (company designators, person first names, etc.). It supports hand-coded grammar rules that make use of part of speech tags, semantic tags added in the gazetteer lookup stage, and if necessary the lexical items themselves. The idea behind supervised learning is to discover discriminative features of named entities by applying machine learning on positive and negative examples taken from large collections of annotated texts. The aim is to automatically generate rules that recognize instances of a certain category entity type based on their features. Supervised learning techniques applied in NEE include Hidden Markov Models (HMM) (Zhou and Su, 2002), Decision Trees (Sekine, 1998), Maximum Entropy Models (Borthwick et al., 1998), Support Vector Machines (Isozaki and Kazawa, 2002), and Conditional Random Fields (CRF) (McCallum and Li, 2003)(Finkel et al., 2005).

Imprecision in information extraction is expected, especially in unstructured text where a lot of noise exists. There is an increasing research interest in more formally handling the uncertainty of the extraction process so that the answers of queries can be associated with correctness indicators. Only recently have information extraction and probabilistic database research been combined for this cause (Gupta, 2006).

Imprecision in information extraction can be represented by associating each extracted field with a probability value. Other methods extend this approach to output multiple possible extractions instead of a single extraction. It is easy to extend probabilistic models like HMM and CRF to return the k highest probability extractions instead of a single most likely one and store them in a probabilistic database (Michelakis et al., 2009). Managing uncertainty in rule-based approaches is more difficult than in statistical ones. In rule-based systems, each rule is associated with a precision value that indicates the percentage of cases where the action associated with that rule is correct. However, there is little work on maintaining probabilities when the extraction is based on many rules, or when the firings of multiple rules overlap. Within this context, (Michelakis et al., 2009) presents a probabilistic framework for managing the uncertainty in rule-based information extraction systems where the uncertainty arises due to the varying precision associated with each rule by producing accurate estimates of probabilities for the extracted annotations. They also capture the interaction between the different rules, as well as the compositional nature of the rules.

2.2 Toponym Disambiguation

According to (Wacholder et al., 1997), there are different kinds of toponym ambiguity. One type is structural ambiguity, where the structure of the tokens forming the name are ambiguous (e.g., is the word "Lake" part of the toponym "Lake Como" or not?). Another type of ambiguity is semantic ambiguity, where the type of the entity being referred to is ambiguous (e.g., is "Paris" a toponym or a girl's name?). A third form of toponym ambiguity is reference ambiguity, where it is unclear to which of several alternatives the toponym actually refers (e.g., does "London" refer to "London, UK" or to "London, Ontario, Canada"?). In this work, we focus on the structural and the reference ambiguities.

Toponym reference disambiguation or resolution is a form of Word Sense Disambiguation (WSD). According to (Buscaldi and Rosso, 2008), existing methods for toponym disambiguation can be classified into three categories: (i) map-based: methods that use an explicit representation of places on a map; (ii) knowledge-based: methods that use external

knowledge sources such as gazetteers, ontologies, or Wikipedia; and (iii) data-driven or supervised: methods that are based on machine learning techniques. An example of a map-based approach is (Smith and Crane, 2001), which aggregates all references for all toponyms in the text onto a grid with weights representing the number of times they appear. References with a distance more than two times the standard deviation away from the centroid of the name are discarded.

Knowledge-based approaches are based on the hypothesis that toponyms appearing together in text are related to each other, and that this relation can be extracted from gazetteers and knowledge bases like Wikipedia. Following this hypothesis, (Rauch et al., 2003) used a toponym's local linguistic context to determine the toponym type (e.g., river, mountain, city) and then filtered out irrelevant references by this type. Another example of a knowledge-based approach is (Overell and Ruger, 2006) which uses Wikipedia to generate co-occurrence models for toponym disambiguation.

Supervised learning approaches use machine learning techniques for disambiguation. (Smith and Mann, 2003) trained a naive Bayes classifier on toponyms with disambiguating cues such as "Nashville, Tennessee" or "Springfield, Massachusetts", and tested it on texts without these clues. Similarly, (Martins et al., 2010) used Hidden Markov Models to annotate toponyms and then applied Support Vector Machines to rank possible disambiguations.

In this paper, we chose to use HMM and CRF to build statistical models for extraction. We developed a clustering-based approach for the toponym disambiguation task. This is described in Section 4.

3 PROBLEM ANALYSIS AND GENERAL APPROACH

The task we focus on is to extract toponyms from EuroCottage holiday home descriptions and use them to infer the country where the holiday property is located. We use this country inference task as a representative example of disambiguating extracted toponyms.

Our initial results from our previous work, where we developed a set of hand-coded grammar rules to extract toponyms, showed that effectiveness of disambiguation is affected by the effectiveness of extraction. We also proved the feasibility of a reverse influence, namely how the disambiguation result can be used to improve extraction by filtering out terms found to be highly ambiguous during disambiguation.

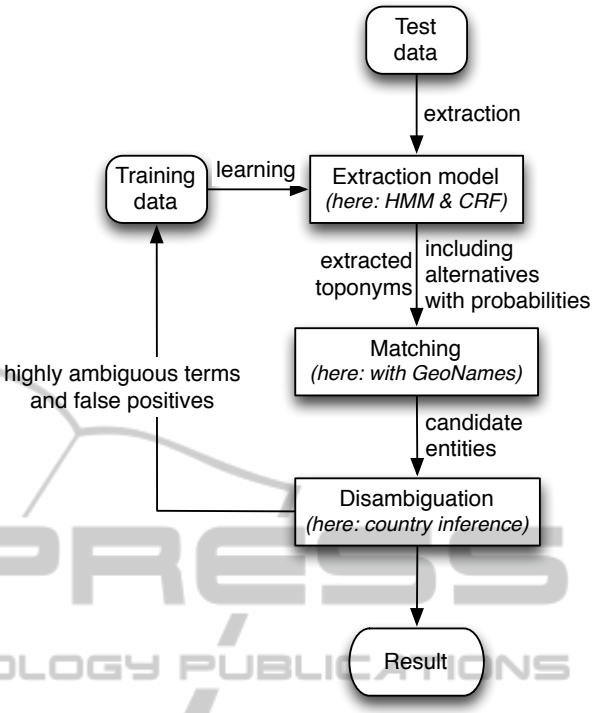


Figure 3: General approach.

One major problem with the hand-coded grammar rules is its "All-or-nothing" behavior. One can only annotate either "Lake Como" or "Como", but not both. Furthermore, hand-coded rules don't provide extraction confidences which we believe to be useful for the disambiguation process. We therefore propose an entity extraction and disambiguation approach based on uncertain annotations. The general approach illustrated in Figure 3 has the following steps:

1. Prepare training data by manually annotating named entities (in our case toponyms) appearing in a subset of documents of sufficient size.
2. Use the training data to build a statistical extraction model.
3. Apply the extraction model on test data and training data. Note that we explicitly allow uncertain and alternative annotations with probabilities.
4. Match the extracted named entities against one or more gazetteers.
5. Use the toponym entity candidates for the disambiguation process (in our case we try to disambiguate the country of the holiday home description).
6. Evaluate the extraction and disambiguation results for the training data and determine a list of highly ambiguous named entities and false positives that affect the disambiguation results. Use them to re-train the extraction model.

7. The steps from 2 to 6 are repeated automatically until there is no improvement any more in either the extraction or the disambiguation.

Note that the reason for including the training data in the process, is to be able to determine false positives in the result. From test data one cannot determine a term to be a false positive, but only to be highly ambiguous.

4 OUR APPROACHES

In this section we illustrate the selected techniques for the extraction and disambiguation processes. We also present our adaptations to enhance the disambiguation by handling uncertainty and the imperfection in the extraction process, and how the extraction and disambiguation processes can reinforce each other iteratively.

4.1 Toponym Extraction

For toponym extraction, we trained two statistical named entity extraction modules[3], one based on Hidden Markov Models (HMM) and one based on Conditional Ramdom Fields (CRF).

4.1.1 HMM Extraction Module

The goal of HMM is to find the optimal tag sequence $T = t_1, t_2, ..., t_n$ for a given word sequence $W = w_1, w_2, ..., w_n$ that maximizes:

$$P(T \mid W) = \frac{P(T)P(W \mid T)}{P(W)} \quad (1)$$

where $P(W)$ is the same for all candidate tag sequences. $P(T)$ is the probability of the named entity (NE) tag. It can be calculated by Markov assumption which states that the probability of a tag depends only on a fixed number of previous NE tags. Here, in this work, we used $n = 4$. So, the probability of a NE tag depends on three previous tags, and then we have,

$$P(T) = P(t_1) \times P(t_2|t_1) \times P(t_3|t_1, t_2) \times P(t_4|t_1, t_2, t_3) \times \ldots \times P(t_n|t_{n-3}, t_{n-2}, t_{n-1}) \quad (2)$$

As the relation between a word and its tag depends on the context of the word, the probability of the current word depends on the tag of the previous word and the tag to be assigned to the current word. So $P(W|T)$ can be calculated as:

$$P(W|T) = P(w_1|t_1) \times P(w_2|t_1, t_2) \times \ldots \times P(w_n|t_{n-1}, t_n) \quad (3)$$

The prior probability $P(t_i|t_{i-3}, t_{i-2}, t_{i-1})$ and the likelihood probability $P(w_i|t_i)$ can be estimated from training data. The optimal sequence of tags can be efficiently found using the Viterbi dynamic programming algorithm (Viterbi, 1967).

4.1.2 CRF Extraction Module

HMMs have difficulty with modeling overlapped, non-independent features of the output part-of-speech tag of the word, the surrounding words, and capitalization patterns. Conditional Random Fields (CRF) can model these overlapping, non-independent features (Wallach, 2004). Here we used a linear chain CRF, the simplest model of CRF.

A linear chain Conditional Random Field defines the conditional probability:

$$P(T \mid W) = \frac{\exp\left(\sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_j f_j(t_{i-1}, t_i, W, i)\right)}{\sum_{t,w} \exp\left(\sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_j f_j(t_{i-1}, t_i, W, i)\right)} \quad (4)$$

where f is set of m feature functions, λ_j is the weight for feature function f_j, and the denominator is a normalization factor that ensures the distribution p sums to 1. This normalization factor is called the *partition function*. The outer summation of the *partition function* is over the exponentially many possible assignments to t and w. For this reason, computing the *partition function* is intractable in general, but much work exists on how to approximate it (Sutton and McCallum, 2011).

The feature functions are the main components of CRF. The general form of a feature function is $f_j(t_{i-1}, t_i, W, i)$, which looks at tag sequence T, the input sequence W, and the current location in the sequence (i).

We used the following set of features for the previous w_{i-1}, the current w_i, and the next word w_{i+1}:

- The tag of the word.
- The position of the word in the sentence.
- The normalization of the word.
- The part of speech tag of the word.
- The shape of the word (Capitalization/Small state, Digits/Characters, etc.).
- The suffix and the prefix of the word.

An example for a feature function which produces a binary value for the current word shape is *Capitalized*:

$$f_i(t_{i-1}, t_i, W, i) = \begin{cases} 1 & \text{if } w_i \text{ is } Capitalized \\ 0 & otherwise \end{cases} \quad (5)$$

[3]We made use of the *lingpipe* toolkit for development: http://alias-i.com/lingpipe

The training process involves finding the optimal values for the parameters λ_j that maximize the conditional probability $P(T \mid W)$. The standard parameter learning approach is to compute the stochastic gradient descent of the *log* of the objective function:

$$\frac{\partial}{\partial \lambda_k} \sum_{i=1}^{n} \log p(t_i|w_i)) - \sum_{j=1}^{m} \frac{\lambda_j^2}{2\sigma^2} \quad (6)$$

where the term $\sum_{j=1}^{m} \frac{\lambda_j^2}{2\sigma^2}$ is a Gaussian prior on λ to regularize the training. In our experiments we used the prior variance σ^2=4. The rest of the derivation for the gradient descent of the objective function can be found in (Wallach, 2004).

4.1.3 Extraction Modes of Operation

We used the extraction models to retrieve sets of annotations in two ways:

- **First-Best.** In this method, we only consider the first most likely set of annotations that maximizes the probability $P(T \mid W)$ for the whole text. This method does not assign a probability for each individual annotation, but only to the whole retrieved set of annotations.
- **N-Best.** This method returns a top-N of possible alternative hypotheses in order of their estimated likelihoods $p(t_i|w_i)$. The confidence scores are assumed to be conditional probabilities of the annotation given an input token. A very low cut-off probability is additionally applied as well. In our experiments, we retrieved the top-25 possible annotations for each document with a cut-off probability of 0.1.

4.2 Toponym Disambiguation

For the toponym disambiguation task, we only select those toponyms annotated by the extraction models that match a reference in GeoNames. We furthermore use a clustering-based approach to disambiguate to which entity an extracted toponym actually refers.

4.2.1 The Clustering Approach

The clustering approach is an unsupervised disambiguation approach based on the assumption that toponyms appearing in same document are likely to refer to locations close to each other *distance-wise*. For our holiday home descriptions, it appears quite safe to assume this. For each toponym t_i, we have, in general, multiple entity candidates. Let $R(t_i) = \{r_{ix} \in$ GeoNames gazetteer$\}$ be the set of reference candidates for toponym t_i. Additionally each reference r_{ix} in GeoNames belongs to a country $Country_j$. By taking one entity candidate for each toponym, we form a cluster. A cluster, hence, is a possible combination of entity candidates, or in other words, one possible entity candidate of the toponyms in the text. In this approach, we consider all possible clusters, compute the average distance between the candidate locations in the cluster, and choose the cluster $Cluster_{min}$ with the lowest average distance. We choose the most often occurring country in $Cluster_{min}$ for disambiguating the country of the document. In effect the abovementioned assumption states that the entities that belong to $Cluster_{min}$ are the true representative entities for the corresponding toponyms as they appeared in the text. Equations 7 through 11 show the steps of the described disambiguation procedure.

$$Clusters = \{\{r_{1x}, r_{2x}, \ldots, r_{mx}\} \mid \forall t_i \in d \bullet r_{ix} \in R(t_i)\} \quad (7)$$

$$Cluster_{min} = \underset{Cluster_k \in Clusters}{\arg\min} \text{ average distance of } Cluster_k \quad (8)$$

$$Countries_{min} = \{Country_j \mid r_{ix} \in Cluster_{min} \wedge r_{ix} \in Country_j\} \quad (9)$$

$$Country_{winner} = \underset{Country_j \in Countries_{min}}{\arg\max} freq(Country_j) \quad (10)$$

where

$$freq(Country_j) = \sum_{i=1}^{n} \begin{cases} 1 & \text{if } r_{ix} \in Country_j \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

4.2.2 Handling Uncertainty of Annotations

Equation 11 gives equal weights to all toponyms. The countries of toponyms with a very low extraction confidence probability are treated equally to toponyms with high confidence; both count fully. We can take the uncertainty in the extraction process into account by adapting Equation 11 to include the confidence of the extracted toponyms.

$$freq(Country_j) = \sum_{i=1}^{n} \begin{cases} p(t_i|w_i) & \text{if } r_{ix} \in Country_j \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

In this way terms which are more likely to be toponyms have a higher contribution in determining the country of the document than less likely ones.

4.3 Improving Certainty of Extraction

In the abovementioned improvement, we make use of the extraction confidence to help the disambiguation to be more robust. However, those probabilities are not accurate and reliable all the time. Some extraction models (like HMM in our experiments) retrieve some false positive toponyms with high confidence probabilities. Moreover, some of these false positives have many entity candidates in many countries according to GeoNames (e.g., the term "Bar" refers to 58 different locations in GeoNames in 25 different countries; see Figure 7). These false positives affect the disambiguation process.

This is where we take advantage of the reinforcement effect. To be more precise, we introduce another class in the extraction model called 'highly ambiguous' and annotate those terms in the training set with this class that (1) are not manually annotated as a toponym already, (2) have a match in GeoNames, and (3) the disambiguation process finds more than τ countries for documents that contain this term, i.e.,

$$\left|\{c \mid \exists d \bullet t_i \in d \wedge c = Country_{winner} \text{ for } d\}\right| \geq \tau \quad (13)$$

The threshold τ can be experimentally and automatically determined (see Section 5.3). The extraction model is subsequently re-trained and the whole process is repeated without any human interference as long as there is improvement in extraction and disambiguation process for the training set. Observe that terms manually annotated as toponym stay annotated as toponyms. Only terms not manually annotated as toponym but for which the extraction model predicts that they are a toponym anyway, are affected. The intention is that the extraction model learns to avoid prediction of certain terms to be toponyms when they appear to have a confusing effect on the disambiguation.

5 EXPERIMENTAL RESULTS

In this section, we present the results of experiments with the presented methods of extraction and disambiguation applied to a collection of holiday properties descriptions. The goal of the experiments is to investigate the influence of using annotation confidence on the disambiguation effectiveness. Another goal is to show how to automatically improve the imperfect extraction model using the outcomes of the disambiguation process and subsequently improving the disambiguation also.

> 2-room apartment 55 m2: living/dining room with 1 sofa bed and satellite-TV, exit to the balcony. 1 room with 2 beds (90 cm, length 190 cm). Open kitchen (4 hotplates, freezer). Bath/bidet/WC. Electric heating. Balcony 8 m2. Facilities: telephone, safe (extra). Terrace Club: Holiday complex, 3 storeys, built in 1995 2.5 km from the centre of **Armacao de Pera**, in a quiet position. For shared use: garden, swimming pool (25 x 12 m, 01.04.-30.09.), paddling pool, children's playground. In the house: reception, restaurant. Laundry (extra). Linen change weekly. Room cleaning 4 times per week. Public parking on the road. Railway station "**Alcantarilha**" 10 km. Please note: There are more similar properties for rent in this same residence. Reception is open 16 hours (0800-2400 hrs). Lounge and reading room, games room. Daily entertainment for adults and children. Bar-swimming pool open in summer. Restaurant with Take Away service. Breakfast buffet, lunch and dinner(to be paid for separately, on site). Trips arranged, entrance to water parks. Car hire. Electric cafetiere to be requested in adavance. Beach football pitch. IMPORTANT: access to the internet in the computer room (extra). The closest beach (350 m) is the "**Sehora da Rocha**", **Playa de Armacao de Pera** 2.5 km. Please note: the urbanisation comprises of eight 4 storey buildings, no lift, with a total of 185 apartments. Bus station in **Armacao de Pera** 4 km.

Figure 4: An example of a EuroCottage holiday home description (toponyms in bold).

5.1 Data Set

The data set we use for our experiments is a collection of traveling agent holiday property descriptions from the EuroCottage portal. The descriptions not only contain information about the property itself and its facilities, but also a description of its location, neighboring cities and opportunities for sightseeing. The data set includes the country of each property which we use to validate our results. Figure 4 shows an example for a holiday property description. The manually annotated toponyms are written in bold.

The data set consists of 1579 property descriptions for which we constructed a ground truth by manually annotating all toponyms. We used the collection in our experiments in two ways:

- **Train_Test Set.** We split the data set into a training set and a validation test set with ratio 2 : 1, and used the training set for building the extraction models and finding the highly ambiguous toponyms, and the test set for a validation of ex-

bath	shop	terrace	shower	at
house	the	all	in	as
they	here	to	table	garage
parking	and	oven	air	gallery
each	a	farm	sauna	sandy

(a) Sample of false positive toponyms extracted by HMM.

north	zoo	west	well	travel
tram	town	tower	sun	sport

(b) Sample of false positive toponyms extracted by CRF.

Figure 5: False positive extracted toponyms.

traction and disambiguation effectiveness against "new and unseen" data.

- **All_Train Set.** We used the whole collection as a training and test set for validating the extraction and the disambiguation results.

The reason behind using the **All_Train set** for traing and testing is that the size of the collection is considered small for NLP tasks. We want to show that the results of the **Train_Test set** can be better if there is enough training data.

5.2 Experiment 1: Effect of Extraction with Confidence Probabilities

The goal of this experiment is to evaluate the effect of allowing uncertainty in the extracted toponyms on the disambiguation results. Both a HMM and a CRF extraction model were trained and evaluated in the two aforementioned ways. Both modes of operation (**First-Best** and **N-Best**) were used for inferring the country of the holiday descriptions as described in Section 4.2. We used the unmodified version of the clustering approach (Equation 11) with the output of **First-Best** method, while we used the modified version (Equation 12) with the output of **N-Best** method to make use of the confidence probabilities assigned to the extracted toponyms.

Results are shown in Table 2. It shows the percentage of holiday home descriptions for which the correct country was successfully inferred.

We can clearly see that the **N-Best** method outperforms the **First-Best** method for both the HMM and the CRF models. This supports our claim that dealing with alternatives along with their confidences yields better results.

5.3 Experiment 2: Effect of Extraction Certainty Enhancement

While examining the results of extraction for both HMM and CRF, we discovered that there were many false positives among the extracted toponyms, i.e., words extracted as a toponym and having a reference in GeoNames, that are in fact not toponyms. Samples of such words are shown in Figures 5(a) and 5(b). These words affect the disambiguation result, if the matching entities in GeoNames belong to many different countries.

We applied the proposed technique introduced in Section 4.3 to reinforce the extraction confidence of true toponyms and to reduce them for highly ambiguous false positive ones. We used the N-Best method for extraction and the modified clustering approach for disambiguation. The best threshold τ for annotating terms as highly ambiguous has been experimentally determined (see section 5.3).

Table 3 shows the results of the disambiguation process using the manually annotated toponyms. Table 5 show the extraction results using the state of the art Stanford named entity recognition model [4]. Stanford is a NEE system based on CRF model which incorporates long-distance information (Finkel et al., 2005). It achieves good performance consistently across different domains. Tables 4 and 6 show the effectiveness of the disambiguation and the extraction processes respectively along iterations of refinement. The "No Filtering" rows show the initial results of disambiguation and extraction before any refinements have been done.

We can see an improvement in HMM extraction and disambiguation results. It starts with lower extraction effectiveness than Stanford model but it outperforms after retraining the model. This support our claim that the reinforcement effect can help imperfect extraction models iteratively. Further analysis and discussion shown in Section 5.5.

Table 2: Effectiveness of the disambiguation process for First-Best and N-Best methods in the extraction phase.

(a) On Train_Test set

	HMM	CRF
First-Best	62.59%	62.84%
N-Best	68.95%	68.19%

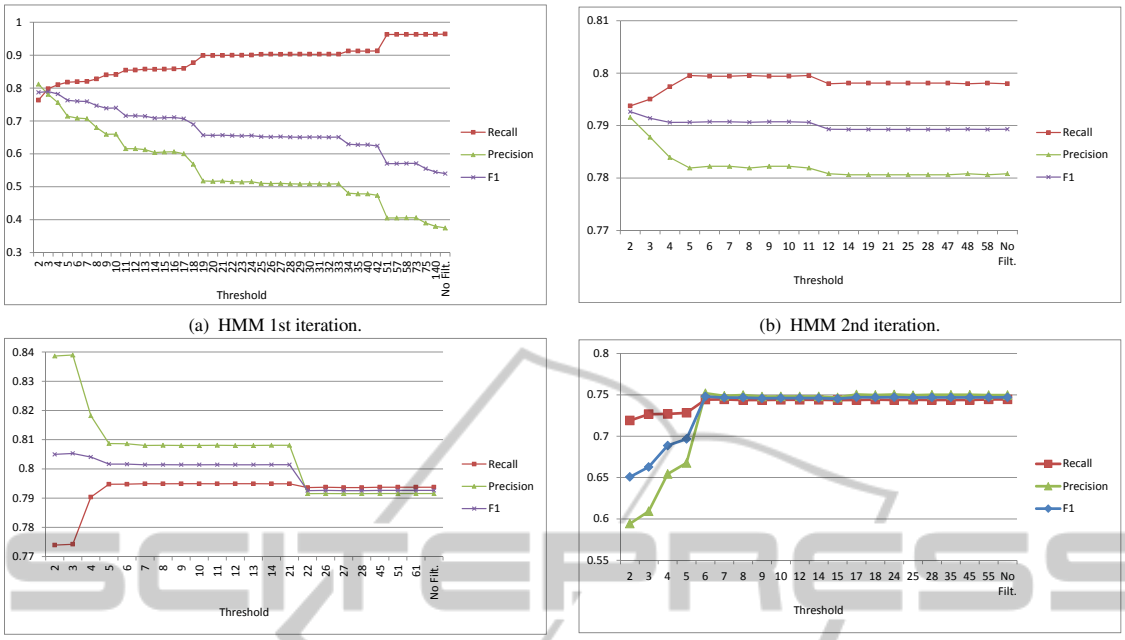
(b) On All_Train set

	HMM	CRF
First-Best	70.7%	70.53%
N-Best	74.68%	73.32%

Table 3: Effectiveness of the disambiguation process using manual annotations.

Train_Test set	All_Train set
79.28%	78.03%

[4] http://nlp.stanford.edu/software/CRF-NER.shtml



(a) HMM 1st iteration.

(b) HMM 2nd iteration.

(c) HMM 3rd iteration.

(d) CRF 1st iteration.

Figure 6: The filtering threshold effect on the extraction effectiveness (On All_Train set)[5]

Table 4: Effectiveness of the disambiguation process after iterative refinement.

(a) On Train_Test set

	HMM	CRF
No Filtering	68.95%	68.19%
1st Iteration	73.28%	68.44%
2nd Iteration	73.53%	68.44%
3rd Iteration	73.53%	-

(b) On All_Train set

	HMM	CRF
No Filtering	74.68%	73.32%
1st Iteration	77.56%	73.32%
2nd Iteration	78.57%	-
3rd Iteration	77.55%	-

Table 5: Effectiveness of the extraction using Stanford NER.

(a) On Train_Test set

	Pre.	Rec.	F1
Stanford NER	0.8385	0.4374	0.5749

(b) On All_Train set

	Pre.	Rec.	F1
Stanford NER	0.8622	0.4365	0.5796

5.4 Experiment 3: Optimal Cutting Threshold

Figures 6(a), 6(b), 6(c) and 6(d) show the effectiveness of the HMM and CRF extraction models

Table 6: Effectiveness of the extraction process after iterative refinement.

(a) On Train_Test set

	HMM		
	Pre.	**Rec.**	**F1**
No Filtering	0.3584	0.8517	0.5045
1st Iteration	0.7667	0.5987	0.6724
2nd Iteration	0.7733	0.5961	0.6732
3rd Iteration	0.7736	0.5958	0.6732

	CRF		
No Filtering	0.6969	0.7136	0.7051
1st Iteration	0.6989	0.7131	0.7059
2nd Iteration	0.6989	0.7131	0.7059
3rd Iteration	-	-	-

(b) On All_Train set

	HMM		
	Pre.	**Rec.**	**F1**
No Filtering	0.3751	0.9640	0.5400
1st Iteration	0.7808	0.7979	0.7893
2nd Iteration	0.7915	0.7937	0.7926
3rd Iteration	0.8389	0.7742	0.8053

	CRF		
No Filtering	0.7496	0.7444	0.7470
1st Iteration	0.7496	0.7444	0.7470
2nd Iteration	-	-	-
3rd Iteration	-	-	-

at first iteration in terms of Precision, Recall, and F1 measures versus the possible thresholds τ. Note that the graphs need to be read from right to left; a lower threshold means more terms being annotated as highly ambiguous. At the far right, no terms are annotated as such anymore, hence this is equivalent to no filtering.

We select the threshold with the highest F1 value. For example, the best threshold value is 3 in figure 6(a). Observe that for HMM, the F1 measure (from right to left) increases, hence a threshold is chosen that improves the extraction effectiveness. It does not do so for CRF, which is prominent cause for the poor improvements we saw earlier for CRF.

5.5 Further Analysis and Discussion

For deep analysis of results, we present in Table 7 detailed results for the property description shown in Figure 4. We have the following observations and thoughts:

- From table 2, we can observe that both HMM and CRF initial models were improved by considering confidence of the extracted toponyms (see Section 5.2). However, for HMM, still many false positives were extracted with high confidence scores in the initial extraction model.
- The initial HMM results showed a very high recall rate with a very low precision. In spite of this our approach managed to improve precision significantly through iterations of refinement. The refinement process is based on removing highly ambiguous toponyms resulting in a slight decrease in recall and an increase in precision. In contrast, CRF started with high precision which could not be improved by the refinement process. Apparently, the CRF approach already aims at achieving high precision at the expense of some recall (see Table 6).
- In table 6 we can see that the precision of the HMM outperforms the precision of CRF after iterations of refinement. This results in achieving better disambiguation results for the HMM over the CRF (see Table 4)
- It can be observed that the highest improvement is achieved on the first iteration. This where most of the false positives and highly ambiguous toponyms are detected and filtered out. In the subsequent iterations, only few new highly ambiguous toponyms appeared and were filtered out (see Table 6).
- It can be seen in Table 7 that initially non-toponym phrases like ".-30.09.)" and "IMPORTANT" were falsely extracted by HMM. These don't have a GeoNames reference, so were not considered in the disambiguation step, nor in the subsequent re-training. Nevertheless they disappeared from the top-*N* annotations. The reason for this behavior is that initially the extraction models were trained on annotating for only one type (toponym), whereas in subsequent iterations they were trained on two types (toponym and 'highly ambiguous non-toponym'). Even though the aforementioned phrases were not included in the re-training, their confidences still fell below the 0.1 cut-off threshold after the 1st iteration. Furthermore, after one iteration the top-25 annotations contained 4 toponym and 21 highly ambiguous annotations.

[5]These graphs are supposed to be discrete, but we present it like this to show the trend of extraction effectiveness against different possible cutting thresholds.

6 CONCLUSIONS AND FUTURE WORK

NEE and NED are inherently imperfect processes that moreover depend on each other. The aim of this paper is to examine and make use of this dependency for the purpose of improving the disambiguation by iteratively enhancing the effectiveness of extraction, and vice versa. We call this mutual improvement, the *reinforcement effect*. Experiments were conducted with a set of holiday home descriptions with the aim to extract and disambiguate toponyms as a representative example of named entities. HMM and CRF statistical approaches were applied for extraction. We compared extraction in two modes, First-Best and N-Best. A clustering approach for disambiguation was applied with the purpose to infer the country of the holiday home from the description.

We examined how handling the uncertainty of extraction influences the effectiveness of disambiguation, and reciprocally, how the result of disambiguation can be used to improve the effectiveness of extraction. The extraction models are automatically retrained after discovering highly ambiguous false positives among the extracted toponyms. This iterative process improves the precision of the extraction. We argue that our approach that is based on uncertain annotation has much potential for making information extraction more robust against ambiguous situations and allowing it to gradually learn. We provide insight into how and why the approach works by means of an

Table 7: Deep analysis for the extraction process of the property shown in Figure 4 (∈: present in GeoNames; #refs: number of references; #ctrs: number of countries).

		GeoNames lookup			Confidence	Disambiguation
	Extracted Toponyms	∈	**#refs**	**#ctrs**	**probability**	**result**
Manually annotated toponyms	Armacao de Pera	√	1	1	-	Correctly Classified
	Alcantarilha	√	1	1	-	
	Sehora da Rocha	×	-	-	-	
	Playa de Armacao de Pera	×	-	-	-	
	Armacao de Pera	√	1	1	-	
Initial HMM model with First-Best extraction method	Balcony 8 m2	×	-	-	-	Misclassified
	Terrace Club	√	1	1	-	
	Armacao de Pera	√	1	1	-	
	.-30.09.)	×	-	-	-	
	Alcantarilha	√	1	1	-	
	Lounge	√	2	2	-	
	Bar	√	58	25	-	
	Car hire	×	-	-	-	
	IMPORTANT	×	-	-	-	
	Sehora da Rocha	×	-	-	-	
	Playa de Armacao de Pera	×	-	-	-	
	Bus	√	15	9	-	
	Armacao de Pera	√	1	1	-	
Initial HMM model with N-Best extraction method	Alcantarilha	√	1	1	1	Correctly Classified
	Sehora da Rocha	×	-	-	1	
	Armacao de Pera	√	1	1	1	
	Playa de Armacao de Pera	×	-	-	0.999849891	
	Bar	√	58	25	0.993387918	
	Bus	√	15	9	0.989665883	
	Armacao de Pera	√	1	1	0.96097006	
	IMPORTANT	×	-	-	0.957129986	
	Lounge	√	2	2	0.916074183	
	Balcony 8 m2	×	-	-	0.877332628	
	Car hire	×	-	-	0.797357377	
	Terrace Club	√	1	1	0.760384949	
	In	√	11	9	0.455276943	
	.-30.09.)	×	-	-	0.397836259	
	.-30.09.	×	-	-	0.368135755	
	.	×	-	-	0.358238066	
	. Car hire	×	-	-	0.165877044	
	adavance.	×	-	-	0.161051997	
HMM model after 1st iteration with N-Best extraction method	Alcantarilha	√	1	1	0.999999999	Correctly Classified
	Sehora da Rocha	×	-	-	0.999999914	
	Armacao de Pera	√	1	1	0.999998522	
	Playa de Armacao de Pera	×	-	-	0.999932808	
Initial CRF model with First-Best extraction method	Armacao	×	-	-	-	Correctly Classified
	Pera	√	2	1	-	
	Alcantarilha	√	1	1	-	
	Sehora da Rocha	×	-	-	-	
	Playa de Armacao de Pera	×	-	-	-	
	Armacao de Pera	√	1	1	-	
Initial CRF model with N-Best extraction method	Alcantarilha	√	1	1	0.999312439	Correctly Classified
	Armacao	×	-	-	0.962067016	
	Pera	√	2	1	0.602834683	
	Trips	√	3	2	0.305478198	
	Bus	√	15	9	0.167311005	
	Lounge	√	2	2	0.133111374	
	Reception	√	1	1	0.105567287	

in-depth analysis of what happens to individual cases during the process.

We claim that this approach can be adapted to suit any kind of named entities. It is just required to develop a mechanism to find highly ambiguous false positives among the extracted named entities. Coherency measures can be used to find highly ambiguous named entities. For future research, we plan to apply and enhance our approach for other types of named entities and other domains. Furthermore, the approach appears to be fully language independent, therefore we like to prove that this is the case and investigate its effect on texts in multiple and mixed languages.

REFERENCES

Borthwick, A., Sterling, J., Agichtein, E., and Grishman, R. (1998). NYU: Description of the MENE named entity system as used in MUC-7. In *Proc. of MUC-7*.

Buscaldi, D. and Rosso, P. (2008). A conceptual density-based approach for the disambiguation of toponyms. *Int'l Journal of Geographical Information Science*, 22(3):301–313.

Finkel, J. R., Grenager, T., and Manning, C. (2005). ncorporating non-local information into information extraction systems by gibbs sampling. In *roceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*, ACL 2005, pages 363–370.

Gaizauskas, R., Wakao, T., Humphreys, K., Cunningham, H., and Wilks, Y. (1995). University of Sheffield: Description of the LaSIE system as used for MUC-6. In *Proc. of MUC-6*, pages 207–220.

Grishman, R. and Sundheim, B. (1996). Message understanding conference - 6: A brief history. In *Proc. of Int'l Conf. on Computational Linguistics*, pages 466–471.

Gupta, R. (2006). Creating probabilistic databases from information extraction models. In *VLDB*, pages 965–976.

Habib, M. B. (2011). Neogeography: The challenge of channelling large and ill-behaved data streams. In *Workshops Proc. of the 27th ICDE 2011*, pages 284–287.

Habib, M. B. and van Keulen, M. (2011). Named entity extraction and disambiguation: The reinforcement effect. In *Proc. of MUD 2011, Seatle, USA*, pages 9–16.

Hobbs, J., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M., and Tyson, M. (1993). Fastus: A system for extracting information from text. In *Proc. of Human Language Technology*, pages 133–137.

Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H., and Wilks, Y. (1998). University of Sheffield: Description of the Lasie-II system as used for MUC-7. In *Proc. of MUC-7*.

Isozaki, H. and Kazawa, H. (2002). Efficient support vector classifiers for named entity recognition. In *Proc. of COLING 2002*, pages 1–7.

Martins, B., Anastácio, I., and Calado, P. (2010). A machine learning approach for resolving place references in text. In *Proc. of AGILE 2010*.

McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proc. of CoNLL 2003*, pages 188–191.

Michelakis, E., Krishnamurthy, R., Haas, P. J., and Vaithyanathan, S. (2009). Uncertainty management in rule-based information extraction systems. In *Proceedings of the 35th SIGMOD international conference on Management of data*, SIGMOD '09, pages 101–114, New York, NY, USA. ACM.

Overell, J. and Ruger, S. (2006). Place disambiguation with co-occurrence models. In *Proc. of CLEF 2006*.

Rauch, E., Bukatin, M., and Baker, K. (2003). A confidence-based framework for disambiguating geographic terms. In *Workshop Proc. of the HLT-NAACL 2003*, pages 50–54.

Sekine, S. (1998). NYU: Description of the Japanese NE system used for MET-2. In *Proc. of MUC-7*.

Smith, D. and Crane, G. (2001). Disambiguating geographic names in a historical digital library. In *Research and Advanced Technology for Digital Libraries*, volume 2163 of *LNCS*, pages 127–136.

Smith, D. and Mann, G. (2003). Bootstrapping toponym classifiers. In *Workshop Proc. of HLT-NAACL 2003*, pages 45–49.

Sutton, C. and McCallum, A. (2011). An introduction to conditional random fields. *Foundations and Trends in Machine Learning*. To appear.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260 – 269.

Wacholder, N., Ravin, Y., and Choi, M. (1997). Disambiguation of proper names in text. In *Proc. of ANLC 1997*, pages 202–208.

Wallach, H. (2004). Conditional random fields: An introduction. Technical Report MS-CIS-04-21, Department of Computer and Information Science, University of Pennsylvania.

Zhou, G. and Su, J. (2002). Named entity recognition using an hmm-based chunk tagger. In *Proc. ACL2002*, pages 473–480.