Anaphora Resolution Using Named Entity and Ontology

Sobha, L
AU-KBC Research Centre
MIT Campus of Anna University
Chennai, India, 600044
sobha@au-kbc.org

Abstract

In this paper we present a knowledge rich approach for resolving anaphors. The present approach shows the need for an in-depth semantic analysis for the proper identification of the antecedent of an anaphor. The semantic disambiguation of the antecedent and anaphor is attained by using a Semantic Disambiguator. The Semantic Disambiguator resolves the issues of animacy and real world identity of the nominals (NP) and thus helps in proposing the most likely candidate antecedent for an anaphor. The base system uses salience factors and salience weight of the candidate NPs for identifying the antecedent from the list of possible candidates for antecedent-hood. The salience weight of an NP is obtained from the salience factors, which are determined by the probability of an NP to be the antecedent on the basis of the grammatical features of the head of NP. We have tested the approach on English ACE[1] data and the results are encouraging.

1 Introduction

Resolution of anaphors plays a vital role in areas such as machine translation, text summarization, and question-answering systems. In machine translating, anaphora must be resolved for languages that mark the gender of pronouns. One major drawback with most current machine translation systems is that the translation usually does not go beyond sentence level, and hence does not deal with discourse understanding successfully. Inter-sentential anaphora resolution would thus be a great assistance to the development of machine translation systems. In the same way the automatic text summarization systems uses a scoring mechanism to identify the most salient sentences. However, the task result is not always guaranteed to be coherent with each other. It creates errors if the selected sentence contains anaphoric expressions. To improve the accuracy of extracting important sentences and also to maintain coherency between sentences, it is essential to solve the problem of anaphoric references in advance. Our earlier research on pronominal anaphora resolution in Tamil (Murthy et al., 2007) showed that the task may be feasible, and depend on the reliability of language specific features such as person, number, gender and case marking. Many resources were lacking for Tamil, and the training and test corpus was limited. The analysis found that identifying the correct antecedent not only depends on the grammatical features of the NP such as subject and direct object, but also depends on the semantic features of the NP. The semantic features are the subcategorization features and the real world identity of the NP. The major bottle-neck for any robust anaphora resolution system is the animacy agreement between the anaphor and the antecedent. The subcategorization features provide the animacy information. Animacy denotes the living entities which can be referred by gender-marked pronouns (*he, she, him, her, his, hers, himself, herself*) in texts. Conventionally, animate entities include people and animals. Since we can hardly obtain the property of animacy with respect to a noun phrase by its surface morphology, we make use of a hierarchical relation, Ontology, for the recognition of animacy of an NP. Consider the following example:

> When council were discussing the weekly bin collection Cllr Lay said "I would cost £669,000 a year to run a weekly service" [£21 a year] Cllr Lay went on to say "There would be a one-off cost of £247,000 to recall residents existing 240 litre bins and a

[1] more information available through the NIST 2008 Automatic Extraction Evaluation (ACE08) http://www.nist.gov/speech/tests/ace/2008/doc/ace08_eval_official_results_20080929.html

> further £700,000 to issue smaller 140-litre bins instead."
>
> Come off it Cllr Lay , WHY must we replace the bins; we could still use them and could if need be, replace when, and as needed. [ACE 2008]

In the above example you have seen that the anaphor "we" refers to "residents". To resolve this we have to identify the real world identity of the NP as well as that of the anaphor. This could be done only if we tag the entity by identifying the named entity of the NP and the anphor as given below. The named entity tags should match or should have a relationship with the anaphor.

> When <GROUP> council </GROUP> were discussing the weekly bin collection <INDIVIDUAL> Cllr Lay </INDIVIDUAL> said "<INDIVIDUAL> I </INDIVIDUAL> would cost £669,000 a year to run a weekly service " [£21 a year] <INDIVIDUAL> Cllr Lay </INDIVIDUAL> went on to say "There would be a one-off cost of £247,000 to recall <GROUP> residents </GROUP> existing 240 litre bins and a further £700,000 to issue smaller 140-litre bins instead."
>
> Come off it <INDIVIDUAL> Cllr Lay </INDIVIDUAL>, WHY must <GROUP> we </GROUP> replace the bins; <GROUP> we </GROUP> could still use <GROUP> them </GROUP> and could if need be, replace when, and as needed.

The paper has the following schema, we present Semantic Disambiguator in the following sections, which has ontology and NER discussed in detail. The anaphora resolution system is described in detail in the next section and the results and discussion at the end.

2 Semantic Disambiguator

The Semantic Diasmbiguator removes the ambiguity related to animacy and real world identity. We use an Ontology and Named Entity in achieving the same. The following sections describe in detail the ontology and the named Entity.

2.1 Ontology

The ontology we use is a language ontology derived from the subcategorization features. The subcategorization features explain the nature of the noun. It includes the features such as $[\pm animate]$, $[\pm concrete]$, $[\pm edible]$ etc. When these features are assigned to nouns, in a sentence, we get more semantic information about the noun in that sentence. The subcategorization features of the noun "airplane" give the characteristics of airplane. It is a non-living entity, physically existing, solid, man-made object. This is a vehicle, which has wheels, and it can fly. The features below give these characteristics.

Airplane : $[-living, +concrete, +movable, +artifact, +solid, +instrument, +vehicle, +wheeled, +avion]$

The subcategorization features of the abstract noun "pain" is given below. This is not a physically existing entity, and this is not virtual. This is a feeling which can be sensed.

Pain : $[-living, -concrete, -virtual, -feature, +sensible, +feeling]$

Some nouns such as fish can have more than one set of features. It can be a living being as well as a food item. The following example illustrates that.

Fish1 : $[+living, +animate, +vertebrate, -mammal, -avion, +fish]$

Fish2 : $[-living, +concrete, +movable, +foodItems, +solid, +animalProd]$

There are totally 104 subcategorization features in our Ontology. A general ontology of nouns is created with the semantic features of the nouns. This is a hierarchical tree, which represents the features of nouns and the subdivisions. It starts with the node entity. This can be sub divided into two: Living and Non-living entities. Living entities is further subdivided into two: Animals and Plants. The divisions in nonliving entities are +Concrete and -Concrete i.e. physically existing and non-existing things. Like this, there are totally 172 nodes in the ontology. This ontology is connected to wordnet using the Wordnet (Miller, 1993) ids for nouns. Wordnet gives different ids for different senses of words. The senses mostly belong to different noun groups (Example: fish belongs to noun.food as well as noun.animal). So, while classifying the nouns in to the ontology nodes itself the sense disambiguation is done. The id of noun.food will go to the node [food items] in the ontology and the id of noun.animal will go to the node fishes which belongs to the [mammal, -avion, fish] group. This classification gives more knowledge to the system. The ontology is a tree diagram with each node having zero or more children. The total number of nouns present in the ontology is $20,000$. We give a sample Ontology tree in figure 1.

2.2 Named Entity

Named Entity Recognition (NER) is the task of identifying and classifying the rigid designators such as person names, place names, organization names etc, in a given document. NER can be visualized as a sequence labeling task and thus can be done with machine learning algorithms supporting sequence labeling task. Our method uses Conditional Random Fields (CRF) (Kudo, 2005; Kudo et al., 2004) for learning from the corpus and tagging new sentences. CRF is a machine learning algorithm suitable for sequence labeling task. CRF extracts features from the training data using the Templates supplied, and learns from the training data the suitable scaling factors for each of those features. We could also supply additional features to CRF and train CRF to learn scaling factors for those features based on training data. The pre-processing required for the training data are Part of Speech tagging and chunking for NP, VP. The features used for CRF training are the word, POS, Dictionaries and Rules. Word and POS features are dynamic features extracted from training data within the window of 5 words with respect to the current token (word) under consideration. The dictionary for training is prepared for each genre as follows. The CRF training is done on the training data using the Initial Dictionary, which we had developed irrespective of the domain. The same training data is tested with CRF testing module and outputs are obtained. From this output, the Named Entities are extracted and added to the Initial Dictionary to form the Final Dictionary for that genre. The training will be done again with this expanded dictionary.

Conditional Random Fields (CRF) (Lafferty et al., 2001) is a machine learning technique, which overcomes the difficulties facing other machine learning techniques like Hidden Markov Model (HMM) (Rabiner, 1989) and Maximum Entropy Markov Model (MEMM) (Berger et al., 1996). HMM does not allow the words in the input sentence to show dependency among each other. MEMM shows a label bias problem because of its stochastic state transition nature. CRF overcomes these problems and performs better than the other two. HMM, MEMM and CRF are suited for sequence labeling task. But only MEMM and CRF allows linguistic rules or conditions to be incorporated into the machine learning algorithm.

Lafferty et al. (2001) define Conditional Random Fields as follows:

> Let $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ be a graph such that $\mathbf{Y} = (\mathbf{Y}_v)_{v \in \mathbf{V}}$, so that $\mathbf{Y}$ is indexed by the vertices of $\mathbf{G}$. Then $(\mathbf{X}, \mathbf{Y})$ is a conditional random field in case, when conditioned on $\mathbf{X}$, the random variables bfY_v obey the Markov property with respect to the graph: $p(\mathbf{Y}_v | \mathbf{X}, \mathbf{Y}_{w, w \neq v}) = p(\mathbf{Y}_v | \mathbf{X}, \mathbf{Y}_{w, w \sim v})$, where $w \sim v$ means that w and v are neighbors in $\mathbf{G}$.

Here $\mathbf{X}$ denotes a sentence and $\mathbf{Y}$ denotes the label sequence. The label sequence y which maximizes the likelihood probability $p_\theta(y|x)$ will be considered as the correct sequence, while testing for new sentence x with CRF model θ. The likelihood probability $p_\theta(y|x)$ is expressed:

$$p_\theta(y|x) \propto$$

$$\exp\Big(\sum_{e \in \mathbf{E}, k} \lambda_k f_k(e, y|e, x) + \sum_{v \in \mathbf{V}, k} \mu_k g_k(v, y|v, x)\Big)$$

where λ_k and μ_k are parameters from CRF model θ and f_k and g_k are the binary feature functions that we need to give for training the CRF model. This is how we integrate linguistic features into machine learning models like CRF.

In the NER task, the sequence of words which forms a sentence or a phrase can be considered as the sequence $\mathbf{x}$ and the sequence formed by named entity label for each word in the sequence $\mathbf{x}$ is the label sequence $\mathbf{y}$. Now, the task of finding $\mathbf{y}$ that best describes $\mathbf{x}$ can be found by maximizing the likelihood probability $p_\theta(y|x)$. Thus, NER task can be considered as a sequence labeling task and hence CRF can be used for NER task. We have used CRF++ (Kudo, 2005; Kudo et al., 2004), an open source toolkit for linear chain CRF. This tool when presented with the attributes extracted from the training data builds a CRF model with the feature template specified.

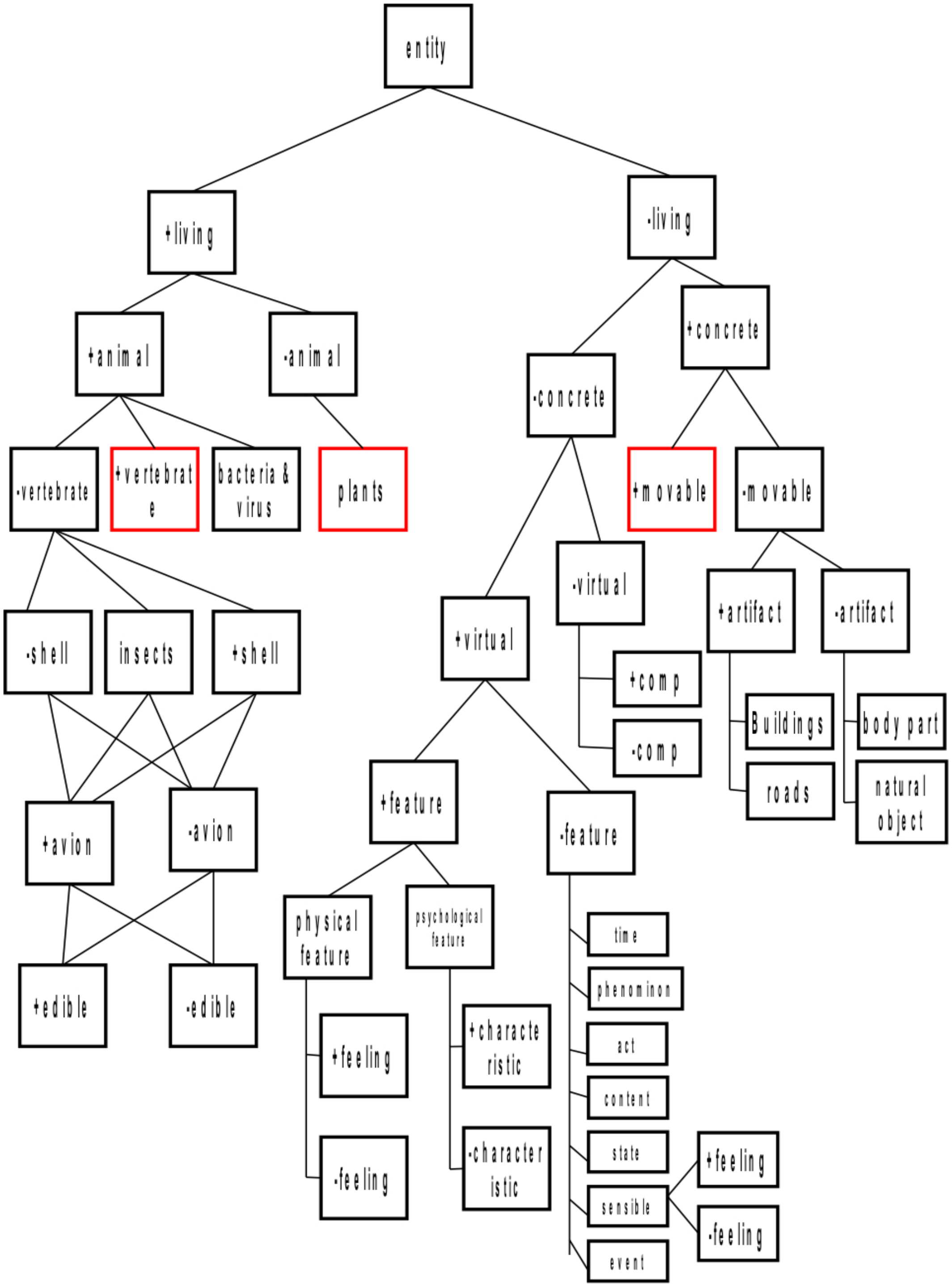

Figure 1: Ontology Tree of Entity

2.3 Presenting training data

Training data will contain nested tagging of named entities. To handle nested tagging and to avoid ambiguities, we isolate the tagset into three subsets, each of which will contain tags from one level in the hierarchy. Now, the training data itself will be presented to CRF as three sets of training data. From this, we will get three CRF models, one for each level of hierarchy. The 94K corpus used are from different domains and they are pre-processed for POS tagging, NP chunking and named entity. There is about 20k named entities in this corpus. The corpus is split into two sets. One forms the training data and the other forms the test data. They consist of 80% and 20% of the total data respectively. CRF is trained with training data. With the model the test data is tagged and the output is evaluated manually. The result is encouraging; precision ∼ 89% and recall 75%.

3 Anaphora Resolution

Anaphora Resolution is the task of identifying the antecedent for an anaphor such as he, she, it, I, you, we etc. Our method uses salience factors for resolving the anaphors. The salience factors are arrived at using linguistic analysis and the salience weights are given according to the preferences each salience factors get. The salience factors used here are similar to factors used by Lappin and Leass(Lappin and Leass, 1994), see also (Lappin and McCord, 1990). The anaphora system we have developed uses the salience factors arrived at by linguistic analysis of the corpus, preference rules and semantic disambiguator. The input to the system is a fully parsed output from FDG parser and the output from NER. The linguistic features taken from FDG parser are Subject, Direct object and Indirect object. The Salience factors and weights are explained below.

3.1 Salience Factors and Weights

The values of salience factors were initially assigned manually, based on linguistic considerations and then fine-tuned through experimentation. The scores are discussed in detail bellow.

a. The current sentence gets a score of 100 and it reduces by 10 for each preceding sentence till it reaches the fifth sentence. The system considers five sentences for identifying the antecedent. Current sentence is the sentence containing the anaphor.

b. The analysis showed that the subject could be the most probable antecedent for the anaphor. The subject noun phrase is given a score of 80.

c. The direct object of a sentence gets a score of 50.

d. The indirect object of a sentence gets a score of 40.

3.2 Agreement features

The NE tags of the anaphor and the NPs are considered for feature agreements. For example if the tag of anaphor is Individual then the NPs with Individual tags alone are considered. We have other feature agreements such as anaphor with Group can have antecedent candidates with Organization. We have also used the pronoun information from the parser for identification of pronouns and pronouns number such as singular or plural. Another rule that is used: Incase if two NPs become the probable candidates with same salience score and the agreement is also same, then the NP which is nearer to the anaphor is considered as the antecedent. We have used the animacy information from the Ontology for checking the animacy between the anaphor and the noun phrases.

4 System Architecture

The system works as follows: The pre-processor processes the input documents for sentence splitting, morphological analysis, POS tagging, NP chunking and Parsing. We compare sentences by sentence with the NER output and the FDG output. The animacy from the Ontology is tagged to the NP. The probable NPs are the NPs that precede the anaphor. The NPs are taken from five sentences above the sentence in which the anaphor occurs. The candidate NPs are identified and checked for the semantic features and named entity matching with the anaphor. The NPs are checked using the agreement features and the NPs which agree with the feature agreement are considered for salience scoring. The salience weight of an NP is the sum of all salience factor values. The NP with the maximum salience weight and agrees in the feature information is considered as the antecedent of the pronoun. The NP which gets the maximum score is considered as the antecedent.

5 Conclusion

Here we present our preliminary results. The corpus we used is the ACE 2008 documents. We have taken 200 documents, from the UseNet domain, which contained 1077 pronouns. The results are highly encouraging. Our NER works with 80% *precision* and 74% *recall*. The system performance was 80% when salience measure was used and an increase of 5% was shown when we used the Semantic Disambiguator. The results reported here are from the ongoing work and the system is not completely evaluated.

Acknowledgement

The Author thank the Department of Science and Technology, Government of India, for funding this research through its *"Women Scientist Award 2004"* scheme.

References

A. L. Berger, V. J. Della Pietra, and S. A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

T. Kudo, K. Yamamoto, and Y. Matsumoto. 2004. Applying conditional random fields to japanese morphological analysis. In *Proc. of EMNLP*.

T. Kudo. 2005. CRF++, an open source system. http://crfpp.sourceforge.net/.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML-2001)*, pages 282–289.

S. Lappin and H. J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

S. Lappin and M. McCord. 1990. Anaphora Resolution in Slot grammar. *Computational Linguistics*, 16(4):197–210.

G. Miller. 1993. Nouns in WordNet: A Lexical Inheritance System. *Journal of Lexicography*, pages 245–264.

Kavi Narayana Murthy, L Sobha, and B Muthukumari. 2007. Pronominal resolution in tamil using machine learning. In Christer Johansson, editor, *Proceedings of the First International Workshop on Anaphora Resolution (WAR-I)*, pages 39–50. Cambridge Scholar Publishing.

L. R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77(2), pages 257–289.