Cross document person name disambiguation using entity profiles

Harish Srinivasan and John Chen and Rohini Srihari
{hsrinivasan,jchen,rohini}@janyainc.com

Janya Inc
1408 Sweet Home Rd, Suit 1
Amherst NY 14228

Abstract. Consolidating entity information spread across multiple documents is a critical problem now with the growing use of large open-domain document sources. Associating every entity in a corpus to a unique entry in a growing knowledge base serves a dual purpose of consolidating (disambiguating) entities as well as to build a rich growing knowledge source containing information about each and every entity accumulated from several documents. With the presence of ambiguous names, use of nominals and aliases, the task of hyper-tagging an entity mentioned in a document to a node in a knowledge base requires the use of context in addition to name matching rules. In this paper, we present an approach that computes a similarity between entities identified in a document with those in the knowledge base using a Vector Space Model utilizing document level entity profiles - Information accumulated for each entity from the entire document. The technique resulted in a TAC evaluation score of 71.9 at the TAC 2009 KBP track. The same technique was also successfully used in obtaining state of the art F-measures (93.95) in disambiguating person names by clustering the similarity values obtained using hierarchical agglomerative clustering.

1 Introduction

One of the primary goals in automatic content extraction from text is identification of the entities mentioned in text. The end goal in this identification is the creation of an ENTITY PROFILE (EP) which contains all the information provided about a particular entity across a set of documents. To create a complete EP, all the individual references to the entity must be identified across the entire corpus and linked together. The latter process is known as COREFERENCE RESOLUTION. Each mention carries with it some additional information about the entity; the EP is a fusion of this content. As new documents flow through the system, new occurrences of entities are detected in these documents. In order to add information from these documents, a system must be able to determine what entity this information is about. The most important piece of information about an entity, for this purpose, is its name. However, although we often think of the name of a thing as a unique designator of that thing, we know that this is not the case in practice. Oftentimes people share the same name, and sometimes a single individual may go by several names. For example, [1] found 35 different individuals named *John Smith* in a collection of New York Times articles from 1996 and 1997. Contrary to this problem, it is also very common to find the same person referred by several variations of the name due to aliases, misspellings, name variants due to transliteration, nicknames, long and short forms. Therefore, in a sufficiently large-scale model, the ability to associate extracted information with the correct individual (known or novel) is critical to the usefulness of that system as a model of the real world. The two different but coupled problems can be formally defined as follows.

1. **Entity Matching:** To group the different entity mentions together despite *not* being mentioned using the same name. Contrary to the above, the variations in the mentions for an entity in a corpus are not only prevalent in person names but also are common to other entity types.
2. **Entity Disambiguation:** To distinguish the different entities despite their being mentioned using the same name. In particular, the study of name disambiguation tends to focus on person names. Location names are also often ambiguous, but the prevalence of ambiguity seems to tail off somewhat for other important entity types such as organizations, facilities, vehicles or military units.

The disambiguation problem is the harder of the two since it is here that coreference resolution needs to be made using plain context. Lately, this has received a lot of interest in the research community. For example, in the 2007 SemEval workshop [2], a competition for disambiguating names from web search was included. ACE 2008 also included cross document coreference resolution, although they focused on the entity matching task. It should be pointed here that although entity matching is overall a relatively easier task, the specific sub problem of identifying aliases is a very difficult one, since the selection of candidates for the search space becomes too large.

2 TAC KBP Entity Linking Task Description

The Knowledge Base Population (KBP) track of Text Analysis Conference(TAC) 2009 formulated the problem of cross document entity consolidation as an *Entity Linking* task described below. *Given a corpus of documents as well as a Knowledge Base (KB) (Wikipedia info-boxes) containing several named entities along with their critical information such as entity type (PER, ORG etc), disambiguating text (Wikipedia text) etc., the goal is to link entities found in a corpus to nodes in the KB. For entries that do not exist in the KB a NIL reference needs to be made.* More specifically, not all the entities in the corpus need be linked but only certain entities of interest given as two tuple queries $< entity\ name >, < document\ id >$. The $< entity\ name >$ corresponds to string with which the entity is mentioned in the document. The task is to output a three tuple $< entity\ name >, < document\ id >, < Knowledge\ Base\ Id >$ such that every entity of interest is linked with the matching entry in the KB. The $Knowledge\ Base\ Id$ is unique identifier by which each entry in the wikipeida infobox is tagged. A *NIL* id is also permissible for those entities that do not correspond to any in the KB.

3 Outline of Approach

Our approach to the *Entity Linking* task is to first index the complete KB to obtain a feature list comprising of bag-of-words (non-stop words) for every entry in the KB. We then process all the documents in the corpus using our IE engine (Semantex) which performs Tokenization, POS tagging, Named Entity Recognition (NER) and within document Co-Reference resolution. The main reason to perform automatic NER even though the entity of interest is specified in the query is to be able to obtain a profile of that entity in that document. In our system, we maintain what could be described as an *entity-oriented* model. The key objects in the model are *entity profiles*, which combine in one place features of the entity, attributes of the entity (links from the entity to a *value*, rather than another entity), relations (to or from another entity), and events that this entity is involved in as a participant. The result of processing a document is a collection of *document-level* entity profiles, which represent all of the information associated with any mention of that entity in the document. The profile also contains the different names (aliases / co-references) with which the entity has been mentioned through out the document and also the entity type (PER,ORG,GPE).

Once the document level entity profiles are obtained using our IE engine, we then select those profiles of interest (entities mentioned in the query) by matching (string match) the query entity name against the entity name in the entity profile. Consider for example a document that contains mentions of $George\ W.\ Bush$ in two different surface forms – $Bush$ and $George\ W.\ Bush$. The entity profile created by processing of the document through our IE engine would list a single profile for the entity $George\ W.\ Bush$ containing $Bush$ as an alias name. Now even if the entity name in the query is just $Bush$, we would be able select the $George\ W.\ Bush$ profile. The key advantage in using entity profiles is that all the alias names listed under the profile of that entity will be used to match against the entity names in the KB – thereby decreasing the chances of not selecting a node in the KB purely based on the entity name in query. Another advantage is that we only need to search through the particular entity type (PER, ORG or GPE) in the KB since the profile of the entity contains this information.

We first employ simple name matching rules described in section 4 to obtain a candidate list of all entities (in the KB) that could potentially be the same as that of entity in the document. This corresponds to selecting all the nodes in the KB whose entity name matches (string match) with any of the alias names in the profile. We term this step as **Entity Matching** as defined previously in section 1. We then use a Vector Space Model (VSM) that employs a rich set of features exracted from the context (contained in the profile), to obtain similarity values between the entity in the document and each and every entity in the candidate list. This step is termed as **Entity Disambiguation** as previously defined in section 1. The entity in the corpus is linked with top ranked candidate if its similarity is greater than a threshold.

4 Entity Matching Model

The entity matching model is a bunch of name matching rules that attempts to retrieve all possible candidate entries from the KB that could potentially be the same entity as of the query entity in the document. The rules are different based on the entity type (PER, ORG or GPE) and they evaluate to *true* or *false* for every trial. All rules are based on case insensitive matches. Two strings are said to match if they satisfy $\frac{d_{edit}\left(S_1, S_2\right)}{min\left(|S_1|, |S_2|\right)} < 0.2$, where $d_{edit}\left(\cdot\right)$ is the levenshtein edit distance between the two strings and $|S|$ is the length of the string. Let us define S_q to be the surface string of the query entity and S_{kb} as the entity in the KB. We now have the following rules.

1. **PER:** Every person name contains *last name* and *first name*. The *last name* is never an empty string. If both *last name* and *first name* are non-empty strings for both S_q and S_{kb}, then they both need to match (as per definition above). If either entity has the *first name* empty, then only the *last name* need match.
2. **ORG:** Organization names tend to be front loaded (eg. Microsoft corp.), i.e. the first token is generally the name of the organization. However there are exceptions like ‘Air France’ etc. We first obtain an inverse document frequency of all the tokens in the entity names for organizations (ORG) in the KB. We only consider matches of strings which are not the most frequently occurring strings (‘non-common’). For a positive match between S_q and S_{kb}, the first ‘non-common’ token in S_q should match (by edit distance) to any token in S_{kb}. For example if $S_q = \mathrm{AirMacau}$ and $S_{kb} = \mathrm{AirFrance}$, then the match is false since the token ‘Air’ is not a ‘non-common’ term.
3. **GPE:** The rules to match location names were the same as that of organizations.

All candidate entities in the KB satisfying the rules are then subjected to disambiguation using context. This model is described next.

5 Disambiguation Model

We employ a Vector Space Model (VSM) to represent the features of the entities. The features for the entities in the KB are just the bag-of-words (non-stop words) in their morphological form extracted from the wikipedia-infobox entry of that entity. For the entities in the document, we employ a rich set of features described in detail below. The basic features consists of the following

1. Summary terms(S): All the non-stop word tokens in the sentence of the entity mention or its coreference are included in their morphological form to the bag-of-words features.
2. Base Noun Phrases (BNP): The non recursive noun phrases in the sentence of entities mention.
3. Document Entities(DE): All the other named entities mentioned in the document.

5.1 Enhancement to features in VSM

Our IE engine is first run on every document in the corpus for named entity recognition and within document coreference resolution. Below we enumerate our modifications to the model.

1. Employing a single bag of words model: We employ a single bag of words model where in we merge the different features together. It was observed in our experiments that this yields better performance. The key reason the separate bag of words model did not perform as well was due to the restriction that terms from one bag of words (say summary sentence terms) are not allowed to match the terms from another bag of words (say DE-document entities). Due to this restriction, common terms that existed across a bag of words between two documents did not count towards the similarity of the two documents.
2. **Profile features (PF):** Our IE system constructs entity profiles, which consolidate features of the entity, attributes of the entity, relations (to or from another entity), and events that this entity is involved in as a participant. The result of processing a document is a collection of document-level entity profiles, which represent all of the information associated with any mention of that entity in the document. All these features associated with the profile are extracted and are stored as two tuple (attribute-value) pairs. The value term in the tuple is then appended to the 'bag of phrases and words'. Table 2 given an example of a profile for an entity named 'John Smith' as extracted by our IE engine. The profile features provides critical information about an entity in a summarized form. In order to extract the profile for an entity, context information from the entire document is utilized. Also event and relation detection is performed to populate these entries in the profile of the entity.

Attribute	Value
PRF_NAM	John Smith
CE_MODIFIERS	Still alive
EVENTS_INVOLVED	Ran into
CE_PER_TITLE	Captain
Ce_Assocication_Entity	Joe Grahame

Table 1. Example of document level entity profile

3. **Topic Model Features (TM):** It was observed that certain pairs of documents had no common terms in their feature space even though, they were about the same ambiguous name. An example is that document 1 contained terms like 'island, bay, water, ship' and document 2 contained terms like 'founder, voyage, and captain'. It is obvious to us that these terms are similar but a naive string matching (VSM model) fails to match these terms at the abstract level. Hence, an expansion of the common noun words in a document was attempted using topic modeling [3]. Using topic modeling, every document is assigned a possible set of topics and every topic is associated with a list of most common words. The following steps were performed to use features from topic model.

(a) The words that were used to learn the topic model were all the nouns in the document along with the terms in the summary sentence. Hence, for each corpus a different topic model was learned due to the difference in the input (words) to the topic model learning algorithm.

(b) The number of topics to learn was set at 50. Once the topic model was learned for each document, the top 10 words with highest *joint probability of word in topic and topic in a document* were chosen. This probability corresponds to the joint probability of word and topic in a document. $P(w, t|D) = P(w|t, D) \times P(t|D) = P(w|t) \times P(t|D)$, where w, t and D are word, topic and document respectively. The last equality in the expression is due to conditional independence of the word and the document, given the topic.

(c) These 10 topic model words are then appended to the existing bag of words and phrases.

4. **Name as a stop word (Nsw):** The ambiguous name in question was included in the stop word list. This is intuitive since the name itself provides no information in resolving the ambiguity as it is present in all the documents. Hence, it was included in the stop word list. It is to be noted here that, for different corpora (each with a different ambiguous name in question), the corresponding names (full name, last name and first name) were added to the stop word list, making it a name specific stop word list.
5. **Prefix matched term frequency (Ptf):** When calculating the term frequency of a particular term in a document, a prefix match was used. e.g. If the term was ‘captain’, and even if only ‘capt’ was present in the document, it is counted towards the term frequency. This modification allows for the possibility of correctly matching commonly used abbreviated words with the corresponding non-abbreviated words.
6. **Log-Transformed Tf-Idf weighting:** The Tf-Idf formulation as used by Bagga and Baldwin is given in equation 1.

$$Sim(S_1, S_2) = \sum_{\text{common terms } t_j} w_{1j} \times w_{2j},$$

$$\text{where} \quad w_{ij} = \frac{\text{tf} \times \ln \frac{N}{\text{df}}}{\sqrt{s_{i1}^2 + s_{i2}^2 + \ldots + s_{in}^2}} \qquad (1)$$

where S_1 and S_2 are the term vectors for which the similarity is to be computed. tf is the frequency of the term t_j in the vector. N is the total number of documents. df is the number of documents in the collection that the term t_j occurs in. The denominator is the cosine normalization.

$$Sim(S_1, S_2) = \sum_{\text{common terms } t_j} w_{1j} \times w_{2j},$$

$$\text{where} \quad w_{ij} = \frac{\ln\left(\text{tf} \times \ln \dfrac{N}{\text{df}}\right)}{\sqrt{s_{i1}^2 + s_{i2}^2 + \ldots + s_{in}^2}} \qquad (2)$$

Our modification to this formulation is given in equation 2.

5.2 Similarity

The cosine-similarity is applied to obtain the similarities between the query entity and all the entries in the candidate list (retrieved from the Entity Matching model). The similarity of the top ranked entry is compared to a threshold (learnt from a manual validation of 100 entries) to decide whether or not to link this KB entry.

5.3 Clustering

This part of the model is not used for experiments in the TAC KBP track but for an alternate set of experiments carried out to benchmark the disambiguation model alone. The disambiguation model can be used stand alone (without any use of KB) to cluster the entities present in a corpus such that each cluster consists of unique entities. Using the above mentioned features and the modified Tf-Idf weighting scheme the cosine-similarity is applied to obtain a # of documents by # of documents similarity matrix. The task now is to cluster the similarity matrix and group documents that mention the same name. Hierarchical agglomerative clustering using single linkage as described previously was used for this purpose. The optimum stop threshold for clustering is then used to compare the clustering results using B-Cubed F-Measure against the key for that corpus. The optimal threshold is defined to be that threshold value where the number of clusters obtained using hierarchical clustering is the same as the number of unique individuals for that given corpus. Typically, in a real world corpus, this information is not known and hence the optimal threshold cannot be found directly. In such a scenario, one uses an annotated data set to learn this threshold and then uses it towards all future clustering. For the sake of comparison between relative importance of features or to compare our results against those published (Chen and Martin [4] and Bagga and Baldwin [1] have also used optimal threshold) previously on the same corpora, it suffices to just use the optimal threshold. The single linkage method yielded the best results among other related techniques such as average linkage, complete linkage, median linkage and weighted linkage.

6 Experiments and Results

We first describe the corpus and evaluation on the TAC KBP track followed by experiments and results for person name disambiguation model.

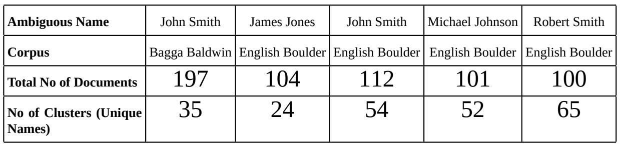
6.1 Experiments and Results in TAC 2009 KBP track

A total of more than $200,000$ entries in the KB were indexed. The number of query entries were 3904. The task was to link every query entry to a node in the KB (NIL link if no appropriate KB entry found). Three different runs were submitted based on changing the threshold of similarity (refer 5.2). The overall precision of the system for the three runs average over all queries were 70.26, 71.08 and 71.08. This corresponds to a median score compared against other evaluations The nature of the task was such that in a majority of times the number of entries in the candidate list retrieved by the Entity

Matching model was just one or nil. This meant that the disambiguation model (where most of our effort was focused) had less significance for the nature of the task. Hence in order to validate the effectiveness of our disambiguation model, we also present in the next section results on a slightly different problem (person name disambiguation) where this model alone is used.

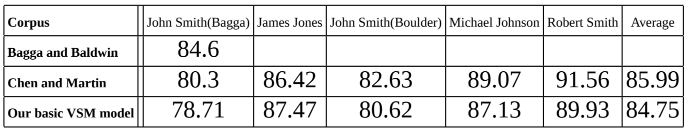
6.2 Experiments and Results for Person Name Disambiguation

The task here is that given a corpus and a ambiguous name (say 'John Smith') to cluster the corpus such that each cluster contains mentions of a unique individual. For this task the disambiguation model alone was used. Two sets of corpora were used for performing experimental evaluations - (i)Bagga Baldwin corpus [1] containing one ambiguous name and (ii)English boulder name corpora containing four sub corpus each corresponding to four different ambiguous names. These together gave a total of five different corpus each one containing a ambiguous name. Table 6.2 summarizes the characteristics of each of the five different corpora. Using the basic VSM model and with no additional features or enhancements, table 3 compares the results obtained by us with that reported by Bagga and Baldwin [1] as well as Chen and Martin [4]. The difference in the performance between the three systems using the same VSM model is due to the difference in the IE engine used and the list of stop words.



Ambiguous Name	John Smith	James Jones	John Smith	Michael Johnson	Robert Smith
Corpus	Bagga Baldwin	English Boulder	English Boulder	English Boulder	English Boulder
Total No of Documents	197	104	112	101	100
No of Clusters (Unique Names)	35	24	54	52	65

Table 2. Corpus description and performance using Bagga Baldwin Model. The F-Measures using Vector Space Model as reported by Bagga and Baldwin, Chen and Martin are included in addition to our implementation of the same. Note that Bagga and Baldwin did not experiment on the English Boulder Name Corpus.



Corpus	John Smith(Bagga)	James Jones	John Smith(Boulder)	Michael Johnson	Robert Smith	Average
Bagga and Baldwin	84.6					
Chen and Martin	80.3	86.42	82.63	89.07	91.56	85.99
Our basic VSM model	78.71	87.47	80.62	87.13	89.93	84.75

Table 3. The F-Measures using Vector Space Model as reported by Bagga and Baldwin, Chen and Martin are included in addition to our implementation of the same.

Table 4 lists the complete set of results with breakdown of the contribution of features as they are added into the complete set. First we show a baseline performance that uses the same set of features as that used by Chen and Martin's best model. The baseline model uses three separate bag of words model, one for each of Summary terms, document entities and base noun phrases and then combines the similarity values using plain average. The difference between our results and those reported by Chen and Martin are due to the difference in the IE engine used, the list of stop words and Chen and Martin's use of Soft TF-IDF weighting scheme. The remaining rows of table 4 use a single bag of words model (all features in the same bag of words) along with the log transformed tf-idf weighting scheme. It can be observed from the table that the addition of features, fine tunings and the use of log-transformed weighting scheme contribute significantly to improve the performance from the baseline. Also, our best model outperforms that reported by Chen and Martin. Model $\mathcal{A}$ + Nsw + Ptf and $\mathcal{A}$ + Nsw + Ptf + TM outperform Chen and Martin's model.

Corpus	John Smith(Bagga)	James Jones	John Smith(Boulder)	Michael Johnson	Robert Smith	Average
No Of Clusters	35	24	54	52	65	
Chen and Martin - Optimal Threshold - S+BNP+DE (Separate bag of words + Soft TF-IDF)	92.02	97.10(28)	91.94(61)	92.55(51)	93.48(78)	93.41
Chen and Martin - Fixed Stop Threshold - S+BNP+DE (Separate bag of words + Soft TF-IDF)	-	96.64	91.31(dev)	90.57(dev)	86.71	93.41
Baseline - S+BNP+DE (Separate bag of words)	84.20(48)	98.11(25)	85.50(62)	90.79(61)	90.37(79)	89.79
Baseline + Log Transformed	93.96(42)	90.54(33)	86.80(71)	89.52(67)	92.66(73)	90.69
Model (Single bag of words + Log Transformed Tf-Idf)						
S+BNP+DE	92.28(50)	95.48(26)	89.50(69)	91.64(49)	92.42(72)	92.26
S+BNP+DE + PF ($\mathcal{A}$)	91.93(47)	98.14(25)	91.46(65)	90.22(57)	92.54(77)	92.85
$\mathcal{A}$ + Nsw	92.77(49)	98.14(25)	90.56(67)	89.85(62)	93.22(70)	92.90
$\mathcal{A}$ + Nsw + Ptf	92.83(49)	98.14(25)	91.24(68)	93.27(55)	94.27(73)	93.95
$\mathcal{A}$ + Nsw + Ptf + TM	**92.62(42)**	**99.03(26)**	**91.49(67)**	**94.01(56)**	**93.03(76)**	**94.03**
$\mathcal{A}$ + Nsw + Ptf + TM (Fixed Stop Threshold)		**94.7(25)**	**89.2(61)**(dev)	**89.92(63)**(dev)	**89.80(67)**	

Table 4. F-measure performance. 'S'-Summary terms, 'PF'-Profile Features, 'BNP'-Base Noun Phrases, 'DE'-Document Entities, $\mathcal{A}$-All features (S+PF+BNP+DE), 'Nsw'-After including the ambiguous Name as a Stop Word, 'Ptf'-Using Prefix matching for calculating Term Frequency, 'TM'-Topic model features. In parenthesis are the number of clusters. In all of the measures, the log-transformed weighting scheme was used along with single linkage clustering. All but the last rows are optimal threshold performances. For the fixed stop threshold, the mean threshold of the 'John Smith (Boulder)' and 'Michael Johnson' were used.

7 Conclusion

The ultimate goal of this research is to be able to update a world model database with consolidated entity information after resolving ambiguities. The nature of the entity linking task in TAC KBP 2009 was such that the entity matching model had greater significance and a median precision score of 71.08 was obtained. High F-measures have been obtained for the task of person name disambiguation validating the effectiveness of the disambiguation model. The extensions to the VSM model described (specifically the profile features and the topic model features) in this paper show an improvement over previously published results. In the future, we plan to tackle the problem of entity disambiguation in combination with that of alias detection and thereby enabling a realization of a system that can correctly (with a high degree of accuracy) consolidate entities from a large corpus.

References

1. Bagga, A., Baldwin, B.: Entity-based cross-document coreferencing using the vector space model. In: Proceedings of COLING-ACL. (1998) 79–85
2. Artiles, J., Gonzalo, J., Sekine, S.: The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. In: Proceedings of the Fourth International Workshop on Semantic Evaluations, Prague (2007)
3. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. In: Journal of machine learning research. Volume 3. (2003) 993–1022
4. Chen, Y., Martin, J.: Towards robust unsupervised personal name disambiguation. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). (2007) 190–198