in Proceedings of the AAAI-2010 Workshop on Collaboratively-built Knowledge Sources and Artificial Intelligence, AAAI Press, July 2010

Approaches for Automatically Enriching Wikipedia

Zareen Syed and Tim Finin

University of Maryland, Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250
{zarsyed1, finin}@umbc.edu

Abstract

We have been exploring the use of Web-derived knowledge bases through the development of Wikitology - a hybrid knowledge base of structured and unstructured information extracted from Wikipedia augmented by RDF data from DBpedia and other Linked Open Data resources. In this paper, we describe approaches that aid in enriching Wikipedia and thus the resources that derive from Wikipedia such as the Wikitology knowledge base, DBpedia, Freebase and Powerset.

Introduction

World knowledge may be available in different forms such as relational databases, triple stores, link graphs, meta-data and free text. People are capable of understanding and reasoning over knowledge represented in different ways and are influenced by different social, contextual and environmental factors. Following a similar model, we can integrate a variety of knowledge sources to produce a single hybrid knowledge base enabling applications to better access and exploit knowledge encoded in different forms. Wikipedia proves to be an invaluable resource for generating a hybrid knowledge base due to the availability and interlinking of structured, semi-structured and un-structured encyclopedic information. However, Wikipedia is designed in a way that facilitates human understanding and contribution by providing interlinking of articles and categories for better browsing and search of information, making the content easily understandable to humans but requiring intelligent approaches for being exploited by applications directly.

Research projects like Cyc (Lenat and Guha, 1989) have produced complex broad coverage knowledge bases but relatively few applications have been built to exploit them. In contrast, the design and development of Wikitology knowledge base has been incremental and has been driven and guided by a variety of applications and approaches that exploit the knowledge available in Wikipedia in different ways.

Wikitology is not unique in using Wikipedia to form the backbone of a knowledge base, see (Suchanek et al., 2008; Wu and Weld, 2008) for other examples. It is unique, however, in integrating knowledge available in different structured and un-structured forms and providing an integrated query interface to applications enabling them to exploit broader types of knowledge resources such as free text, relational tables, link graphs and triples (Figure 1). In general, each article in Wikipedia represents a concept or individual. There are many advantages in deriving a knowledge base from Wikipedia such as having broad coverage (over three million articles); representing a consensus based concept space developed and kept current by a diverse collection of people; comprising high quality of content (Hu et al. 2007) and being available in multiple languages. The intended meaning of the pages, as concepts, is self evident to humans, who can read the text and examine the images on the pages.

In this paper, we describe our previous approaches as well as our novel approach to discovering infobox like

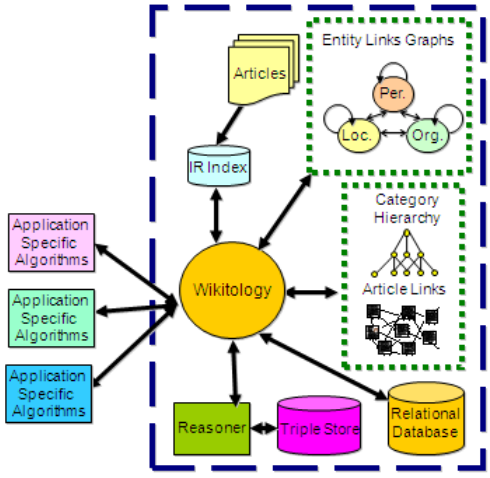

Figure 1. Wikitology is a hybrid knowledge base storing information in structured and unstructured forms and reasoning over it using a variety of techniques.

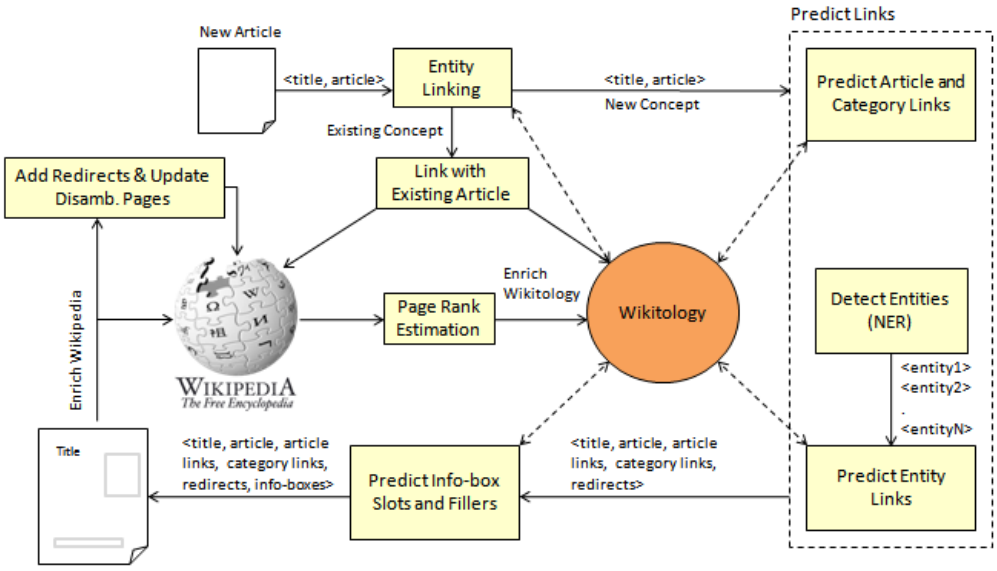

Figure 2. Approach for adding new articles to Wikipedia showing different Wikitology based approaches contributing towards automatically enriching Wikipedia and hence the Wikitology Knowledge Base itself.

slots and fillers that contribute to different steps in the unified framework for automatically enriching Wikipedia and hence the Wikitology knowledge base with articles on new concepts.

The proposed unified framework may be suitable for adding to or augmenting Wikipedia articles based on articles found in other encyclopedias such as Britannica, Microsoft Encarta or Hutchinson. This will in turn result in enriching resources that are derived from Wikipedia such as DBpedia (Auer et al. 2007), Freebase (Bollacker et al. 2007), Powerset (www.powerset.com) and linked open data collections (Bizer, 2009).

Enriching Wikipedia and Wikitology

Since Wikitology is derived from Wikipedia, automatically enriching Wikipedia directly contributes to enriching the Wikitology knowledge base. Wikitology has been successfully employed for a variety of tasks such as document concept prediction (Syed et al. 2008), the cross document coreference resolution task (Finin et al. 2009) defined in Automatic Content Extraction (ACE) (Strassel et al. 2008), and the entity linking task (Finin and Syed, 2010) defined as a part of 2009 Text Analysis Conference Knowledge Base Population Track (McNamee and Dang, 2009).

In addition to our existing work, we have also developed specific approaches aimed at automatically enriching the Wikitology KB by unsupervised discovery of infobox slots and fillers, generating disambiguation trees for entities and estimating the PageRank of Wikipedia articles to serve as a measure of popularity. The different Wikitology based approaches combined together can contribute to a number of steps in a broader unified framework for adding an article on a new concept in Wikipedia and hence in the Wikitology KB (Figure 2).

Automatically adding a text article on a new concept to Wikipedia requires identifying appropriate categories, inter-article links and infobox properties and values. There might also be a need to associate one or more redirect pages to the new article. Furthermore, if the concept title is ambiguous with other existing concepts, it also requires updating the relevant disambiguation page. The different steps in the framework generate the necessary structured data that is needed to construct a typical Wikipedia article and integrate it into existing Wikipedia structure.

The input to our framework is a text article on a concept along with the concept title. We use our entity linking approach (Finin and Syed, 2010) to predict if an article on the same concept already exists in Wikipedia. If the article is predicted to be a new concept, we follow on to the next step and predict categories and inter-article links for the new article using our document concept prediction approach (Syed et al. 2008). We can use a named entity recognition (NER) system to detect named entities in the new article and explicitly link those named entities mentioned in text of the article to the right Wikipedia entity articles using our entity linking approach (Finin and Syed, 2010). We can also generate redirects for the new concept as a side effect of entity linking. The next step in our approach exploits the inter-article links for discovering infobox like structured data for the new article. The new article with categories, inter-article links and infobox slots and fillers along with redirects if any, can then be added to Wikipedia. If the concept title is ambiguous, the relevant disambiguation pages can be updated as well.

Match	Accuracy
Positive Match (Entity exists in Wikipedia)	81.1%
Negative Match (Entity doesn't exist in Wikipedia)	78.7%

Table 1. Accuracy obtained for positive and negative entity matches using Wikitology

Prediction	P	Avg. P	R	F
Category Prediction	0.62	0.98	0.94	0.75
Link Prediction	0.59	0.94	0.88	0.67

Table 2. Evaluation for Category and Link Prediction using Document Concept Prediction approach

Entity Type	Accuracy
Persons	94.9%
Locations	85.0%
Organizations	82.8%

Table 3. Evaluation for Link prediction to different types of entities using Entity Linking Approach

Adding new Article

Our approach is to check if an article already exists on that concept. In order to add a new article to Wikipedia, the first step uses our existing entity linking approach (Finin and Syed, 2010) to determine this. Given an entity or concept mention string (title) and an article (free text), our approach constructs a specialized query to Wikitology and retrieves the top ranked concept returned by Wikitology. If the score for the top ranked concept is above a threshold we predict that concept as the link to the right entity in Wikipedia, otherwise we predict the input concept as a "new concept". For the details related to the approach please see Finin and Syed (2010). We evaluated the approach using 3000 Wikinews (http://en.wikinews.org) articles mentioning persons, locations and organizations, the accuracy obtained for a positive match i.e. concept exists in Wikipedia and a negative match i.e. a new concept, are given in Table 1. In case the concept exists we can update the existing article and add a reference to the new article in the "external links" section and refrain from adding duplicate concepts. In case the entity or concept doesn't exist we can follow the steps to adding the article on the new concept to Wikipedia and the Wikitology KB.

Predicting Categories and Inter-article Links

The second step in our approach predicts the categories and the inter-article links for the new article. We use our existing approach for document concept prediction to predict the generalized concepts i.e. categories as well as specialized concepts i.e. related Wikipedia articles for a new document (Syed et al. 2008).

For a given document as input we retrieve the top N similar Wikipedia articles from the Wikitology index and use them as seed nodes for spreading activation algorithm on the page links graph to predict specialized concepts related to the input document. The categories associated with the top N similar articles are used as seed nodes for spreading activation on the category links graph for predicting generalized concepts or categories for the input document.

We evaluated the approach by predicting the categories and article links of existing Wikipedia articles and compared them with the ground truth using measures for precision, average precisions, recall and f-measure. We observed that the greater the average cosine similarity between the new document and the retrieved top N Wikipedia articles, the better the prediction. The average similarity with the top N retrieved Wikipedia articles can also be used as a weight on the links to represent the confidence in accuracy of the predictions. The evaluation results for articles with at least 0.5 average cosine similarity with the top N retrieved Wikipedia articles are given in Table 2. For more details regarding the approach and evaluation, please see Syed et al. (2008).

While our concept prediction approach can predict related concepts/articles to link, we also have specialized approaches for linking different types of entities to the relevant Wikipedia articles on entities (Finin and Syed, 2010). We evaluated our entity linking approach for entities mentioned in Wikinews articles. The accuracy obtained for different types of entities is given in Table 3.

We can use any existing named entity recognition (NER) system to identify entities (e.g., persons, locations and organizations) mentioned in the new article. Given the entity mention and the article with the mention, we can use our entity linking approach to link to the right Wikipedia article. We can also identify the redirects to articles as a side effect of entity linking. The different entity mentions that link to the same entity can serve as redirects to that entity provided they are unique and do not exist as redirects for any existing concepts.

Our document concept prediction approach predicts related concepts however, does not define where to anchor the links in the text of the article. The predicted concepts can be added to the "See also" list in the article. Whereas, using an NER system and entity linking approach we can anchor the links at specific entity mention strings in the new article.

Discovering Infobox Slots and Fillers

In addition to exploiting our existing approaches, we have developed a novel unsupervised and unrestricted approach to discovering info-box like slots and fillers by exploiting the inter-article links within Wikipedia which represent relations between concepts. In our approach we consider the linked concepts as candidate fillers for slots related to the primary article/concept. There are several cases where the filler is subsumed by the slot label for example, the infobox present in the article on "Michael_Jackson" mentions *pop*, *rock* and *soul* as fillers for the slot "Genre" and all three of these are a type of Genre. Based on this observation, we discover and exploit "ISA"' relation between

fillers (linked concepts) and WordNet nodes to serve as candidate slot labels.

We first identify and rank candidate slots and fillers for an entity and then classify it based on ranked slots and re-rank the slots based on classification. We further perform slot selection to discard any irrelevant slots.

Discovering Candidate Slots and Fillers. We discover and exploit "ISA"' relation between fillers (linked concepts) and WordNet nodes to serve as candidate slot labels. In order to find "ISA" relation between a concept and a WordNet synset we use manually created mappings by DBpedia, which links about 467,000 articles to synsets, to map any remaining concepts we use the automatically generated mappings available between WordNet synsets and Wikipedia categories (Ponzetto and Navigli, 2009). A single Wikipedia article might have multiple categories associated with it and therefore multiple WordNet synsets. In order to select an individual WordNet synset to represent the type of the concept we use the following two heuristics.

The first heuristic attempts to extract a category label from the first sentence of the article, which usually defines the concept. As an example, consider querying for "Talis", which leads one (via a disambiguation) to a page for "Talis Group", which begins with the sentence:

> "**Talis Group Ltd.** is a software company based in Solihull near Birmingham, England that develops a Semantic Web application platform and a suite of applications for the education, research and library sectors."

We extract a category label from the first sentence using patterns based on part-of-speech tags similar to Kazama and Torisawa (2007).

The second heuristic tries to match a WordNet synset to the category label. We assign the WordNet synset if any of the words in the synset matches with the extracted category label. We repeat the process with hypernyms and hyponyms of the synset up to three levels.

Slot Ranking. All the slots discovered using outgoing links might not be meaningful. We have developed techniques for ranking and selecting slots. Our approach is based on the observation that entities of the same type have common slots. For example, there is a set of slots common for musical artists whereas, a different set is common for basketball players. This is also evident in the Wikipedia infobox templates based on classes.

In case of entities of a particular type, it is a common observation that there are a set of slots that are generalized, i.e., they are common across all entities of that type. For example, for persons common slots are name, born, spouse, etc. There are also sets of specialized slots which are generally related to sub-type of entity, in case of people it is often related to their profession. For example, the slots for basketball players have information related to basketball related activities and Musical Artists have slots with music related activities. The slots for "Michael_Jordan" include Professional Team(s), NBA Draft, Position(s) and slots for "Michael_Jackson" include Genres, Instruments and Labels.

Another observation is that people engaged in a particular profession tend to be linked to others within the same profession. Hence the maxim "A man is known by the company he keeps.", for example, basketball players are linked to other basketball players and politicians are linked to other politicians. We rank the slots based on the number of linked entities of the same type (in our case "persons") having the same slots. We generated a list of person articles in Wikipedia by getting all Wikipedia articles under the Person type in Freebase. We randomly select up to 25 linked persons (which also link back) and extract their candidate slots and vote for a slot based on the number of times it appears as a slot in a linked person normalized by the number of linked persons to assign a slot score.

Entity Classification and Slot Re-ranking. We use the ranked candidate slots to classify entities and then further rank the slots based on number of times they appear among the entities in the cluster. We use complete link clustering using the cosine similarity between tf.idf weighted slot vectors, where the slot score represents the term frequency component and the inverse document frequency is based on the number of times the slot appears in different individuals.

$$sim_{slot}(p_i, p_j) = \cos(slot(p_i), slot(p_j))$$

We also collapsed location expressing slots (country, county, state, district, island etc.) into the slot labeled location by generating a list of location words from WordNet as these slots were causing the persons related to same type of geographical location to cluster together.

After clustering we re-score the slots based on number of times they appear among the individuals in the cluster normalized by the cluster size. The output of clustering is a vector of scored slots associated with each cluster or class.

Slot Selection. To filter out any irrelevant slots we have an approach for slot selection. Our intuition is that specialized slots or attributes for a particular entity type should be somehow related to each other. For example, we would expect attributes like *league*, *season* and *team* for basketball players and *genre*, *label, song* and *album* for musical artists. If an attribute like *album* appears for basketball players it should be discarded as it is not related to other attributes. We adopted a clustering approach for finding attributes that are related to each other. For each pair of attributes in the slot vector we computed a similarity score based on how many times the two attribute labels appear together in Wikipedia person articles within a distance of 100 words as compared to the number of times they appear in total and weigh it using weights of the individual attributes in the slot vector. This metric is captured in the following equation where *Df* is the document frequency and *wt* is the attribute weight.

Slot	Score	Fillers Example
Musician	1.00	ray_charles, sam_cooke ...
Album	0.99	bad_(album), ...
Location	0.97	gary,_indiana, chicago, …
Music_genre	0.90	pop_music, soul_music, ...
Label	0.79	a&m_records, epic_records, ...
Phonograph_ record	0.67	give_in_to_me, this_place_hotel …
Act	0.59	singing
Movie	0.46	moonwalker …
Company	0.43	war_child_(charity), …
Actor	0.41	stan_winston, eddie_murphy,
Singer	0.40	britney_spears, …
Magazine	0.29	entertainment_weekly,…
Writing_style	0.27	hip_hop_music
Group	0.21	'n_sync, RIAA
Song	0.20	d.s._(song) …

Table 4. Fifteen slots were discovered for musician Michael Jackson along with scores and example fillers.

$$sim_{attr}(a_i, a_j) = wt(a_i) \times wt(a_j) \times \frac{Df(a_i, a_j, 100)}{Df(a_i) + Df(a_j)}$$

We did some initial experiments using single link and complete link and found single link to be more appropriate for slot selection. We got clusters at a partition distance of 0.9 and selected the largest cluster from the set of clusters. In addition to that we also added any attributes exceeding a 0.4 score into the set of selected attributes. Selected ranked slots for Michael Jackson are given in Table 4.

Experiments and Evaluation. For our experiments and evaluation we used the Wikipedia dump from March 2008 and the DBpedia infobox ontology created from Wikipedia infoboxes using hand-generated mappings (Auer et al., 2007). The Person class is a direct subclass of the owl:Thing class and has 21 immediate sub-classes and 36 subclasses at level two. We used the persons in different classes at level two to generate data sets for experiments.

There are several articles in Wikipedia that are very small and have very few out-links and in-links. Our approach is based on the out-links and availability of information about different related things on the article, therefore, in order to avoid data sparseness, we randomly select articles with greater than 100 in-links and out-links, at least 5KB page length and having at least five links to entities of the same type that link back (in our case persons). We selected all the sub-classes of Person class at level 2. We randomly selected 200 person articles using the same criteria already defined to avoid data sparseness. We discarded classes for which we got fewer than 20 articles. Our final data set comprised 28 classes and 3810 articles.

An initial comparison of ranked slots features with other feature sets extracted from Wikipedia showed that the ranked slots features gave the highest accuracy w.r.t F-measure (i.e. 0.73) for entity classification when compared with words in first sentence (0.12), categories (0.60) and WordNet nodes (0.12) associated with articles.

Properties	Accuracy
automobile_race, championship, expressive_style, fictional_character, label, racetrack, team_sport, writing_style	100%
academic_degree, album, book, contest, election, league, phonograph_record, race, tournament, award, movie, novel, school, season, serial, song	>= 90%
car, church, game, musical_instrument, show, sport, stadium, broadcast, telecast	>= 80%
hockey_league, music_genre, trophy	>= 70%
university, character, disease	>= 60%
magazine, team, baseball_club, club, party, captain, coach	< 60%
Avg. Accuracy:	81%

Table 5. Manual evaluation of discovered properties for different entity classes.

We used our ranked slots tf.idf feature set and ran complete link clustering algorithm producing clusters at partition distance of 0.8. We re-scored the slots based on the number of times they appeared in the cluster members normalized by the cluster size. We applied slot selection over the re-scored slots for each cluster. In order to evaluate our slots and fillers we mapped each cluster to a DBpedia class based on the maximum number of members of a particular DBpedia class in our cluster. In total our approach predicted 124 unique properties corresponding to different classes and out of 124 properties we were able to find 46 properties that existed in either DBpedia ontology or Freebase for the corresponding class.

We initially tried to evaluate the discovered slots by comparing them with DBpedia ontology and Freebase however, we were able to find an overlap in the subject and object pairs for very few properties. Therefore we decided to evaluate the pairs manually. We randomly selected 20 subject object pairs for each of the 46 properties from the corresponding classes and manually evaluated if the relation was correct by consulting the corresponding Wikipedia articles (Table 5).

The highest accuracy of 100% was obtained for the following eight properties: *automobile_race, championship, expressive_style, label, racetrack, team_sport, fictional_character, writing_style*. In total 33 properties had accuracy of greater than or equal to 80%. For few properties the accuracy was 60% or below which are *coach, captain, baseball_club, club, party, team and magazine*. The average accuracy for the 46 relations was 81%.

We observed that certain properties such as *spouse, predecessor, successor,* etc. require more contextual information and are not directly evident in the link structure. While our work has mostly experimented with person entities, the approach can be applied to other types as well. For example, we were able to discover *software* as a candidate slot for companies like Microsoft, Google and Yahoo!, which appeared among the top three ranked slots using our slot ranking scheme and corresponds to the *products* slot in the infoboxes of these companies. We also observed that our approach discovered certain slots that may not be ap-

```
Profession = musician: Michael_Jackson
Profession = 0
|  Nationality = english: Michael_Jackson_(footballer)
|  Nationality = 0: Michael_Jackson_(Anglican_bishop)
Profession = guitarist: Michael_Gregory_(jazz_guitarist)
Profession = sheriff: Michael_A._Jackson_(sheriff)
Profession = journalist: Michael_Jackson_(journalist)
Profession = player: Michael_Jackson_(basketball)
Profession = executive: Michael_Jackson_(television_executive)
Profession = writer: Michael_Jackson_(writer)
Profession = professor: Michael_Jackson_(anthropologist)
Profession = footballer: Michael_Jackson_(rugby_league)
Profession = scientist: Michael_A._Jackson
Profession = soldier: Michael_Jackson_(American_Revolution)
Profession = actor
|  Nationality = english: Michael_J._Jackson_(actor)
|  Nationality = canadian: Michael_Jackson_(actor)
```

Figure 3. This automatically compiled disambiguation tree helps link the mention "Michael Jackson" to the appropriate Wikitology entity.

propriate to include in the infoboxes such as *musician* for Michael Jackson which has other musicians related to Michael Jackson as fillers, however, such structured information may be useful for answering certain queries such as *which musicians are related to Michael Jackson?*.

Generating Disambiguation Trees

Wikipedia has special, manually created disambiguation pages for sets of entities with identical or similar names. We have developed an approach to disambiguate mentions that refer to a particular Wikipedia entity. Not all confusable entities have such disambiguation pages and an automated process for creating them could both contribute to Wikipedia and also support entity linking and disambiguation.

Different sources within Wikipedia can be exploited to extract confusable or ambiguous entity mentions for generating disambiguation pages automatically. Cucerzan (2007) used the titles of entity pages, redirection pages, existing disambiguation pages and surface form of text associated with links to entity pages in other Wikipedia articles for developing a named entity disambiguation system. The same sources can be used to identify ambiguous entity mentions in Wikipedia and a disambiguation page can be created for each ambiguous mention.

We have developed an initial prototype to create disambiguation pages for people (Finin and Syed, 2010). We modeled this problem as a multiple class classification problem where each person with a similar name is considered an individual class. We extract nouns and adjectives from the first sentence of the person articles and use them to construct a decision tree. Most commonly, the nodes that were selected to split on referred to either the person's nationality or profession.

We enhanced our approach by using a domain model (Garera and Yarowsky, 2009) of nationalities and professions constructed from Wikipedia's list pages "list_of_nationalities" and "list_of_professions". These were used to extract the nationality and profession of persons by selecting nouns and adjectives as features that appeared in the first sentence and were in one of the domain models. Using the nationality and profession as features, we constructed a decision tree for different Persons having a confusable name. When we were not able to extract a profession or nationality of the entity from the first sentence, we gave that feature a value of "0". We refer to these decision trees that help in disambiguating entities as *disambiguation trees*.

Features	Accuracy	± 1 Accuracy
All features	53.14	93.78
All - Page Length	53.91	95.30
All - InLinks	51.46	93.23
All - OutLinks	54.81	95.34
In-links	54.03	95.69
Page Length	49.95	94.20
Out-links	49.46	94.13

Table 6. An analysis of our features revealed that only using the number of in-links provided our best approximation for Google's PageRank metric for a Wikipedia article.

We have constructed several disambiguation trees for different sets of persons having the same name. An example of the disambiguation tree constructed for people having the name "Michael Jackson", is shown in Figure 3. Out of 21 different people named "Michael Jackson" we were able to disambiguate fifteen of them using just the profession and nationality features.

For locations with confusable names, we have observed that another location can help in disambiguating it. For example, the disambiguation page on "Springfield" refers to 64 different place names in Wikipedia. In almost all cases it is disambiguated by using another place name or location which is usually mentioned in the first sentence of the respective article. Therefore, disambiguation trees for locations can be generated using other locations mentioned in the first sentence.

In general the nouns and adjectives in the first sentence of articles on concepts can provide discriminatory features for disambiguating in majority of the cases. In order to disambiguate different entity types such as persons, locations or organizations an NER system could be used on sentences having inline links to that particular entity in Wikipedia and their entity type could be detected. We plan to evaluate our approach for generating disambiguation trees on a large data set in the future.

Estimating Popularity

Earlier work on Entity Linking task (McNamee et al. 2009) identified Google page rank as an important feature for linking entities mentioned in text to Wikipedia entities. We

have developed an approach for approximating the page rank for Wikipedia articles. We took 4296 random articles from Wikipedia and queried Google for their PageRank. We considered the in-links, out-links and page length as features for the classifier and the PageRank as the class label. We classified the Wikipedia articles into one of the PageRank classes using the three classification algorithms, i.e. SVM, Decision Trees and Naïve Bayes. The highest accuracy was obtained by using the Decision Tree classifier. We computed the accuracy using 10 fold cross validation. We also compared different combinations of features. We re-evaluated the accuracy and considered a prediction to be correct if it was within a range of +/-1 of the actual PageRank (Table 6). In case of accuracy, in-links + page length gave the best results (54.81%) which were followed by in-links only (54.03%). When the Accuracy was computed considering a prediction to be correct if it is within +/- 1 range of the PageRank, the highest accuracy was obtained by in-links only (95.69%). We have employed the predicted PageRanks for linking entities mentioned in tables to Wikipedia entities and for predicting the column headers (Syed et al. 2010).

Related Work

To our knowledge there doesn't exist a single complete system that supports all steps of adding a new article automatically to Wikipedia. However, there are a few systems that support individual steps of adding a new article and can complement the different steps in our proposed framework. We discuss those systems below.

Schonhofen (2006) developed an approach for identifying document topics using Wikipedia category network and evaluated the approach by predicting categories of Wikipedia articles. Wikify (Mihalcea and Csomai, 2007) is an approach that annotates any given text with links to Wikipedia using keyword extraction and word sense disambiguation. Milne and Witten (2008) use a machine learning approach for cross referencing documents within Wikipedia. Both systems can complement our approach for discovering inter-article links for the new article.

Suchanek et al. (2008) used Wikipedia categories and infoboxes to extract relations by applying specialized heuristics for each relation type and incorporated the relations in their YAGO ontology. Kylin (Weld et al. 2008) generated infoboxes for articles by learning from existing infoboxes. KOG (Wu and Weld, 2008) automatically refined the Wikipedia infobox ontology and integrated Wikipedia's infobox-class schemata with WordNet. All the three systems YAGO, Kylin and KOG rely on relations present in the infoboxes. Our infobox prediction approach can complement these approaches by discovering new relations evident in inter-article links in Wikipedia. For example, we can add slots like *songs* and *albums* to the infobox schema for Musical Artists, *movies* for the Actors infobox schema, and *party* for the Politicians schema. We already use the WordNet nodes for representing slots, hence, it eliminates the need for several refinement steps taken by KOG to refine the infoboxes.

Conclusions and Future Work

We have presented several approaches that can contribute to different steps in the unified framework for enriching Wikipedia and hence the Wikitology KB by adding articles on new concepts. Our proposed unified framework may be suitable for adding articles in other general knowledge encyclopedias such as Britannica, Microsoft Encarta and Hutchinson Encyclopedia as well as articles in domain specific encyclopedias to Wikipedia for the concepts that do not already exist in Wikipedia. This will in turn result in enriching resources that are derived from Wikipedia such as DBpedia, Freebase, Powerset and linked open data collections. We plan to refine and extend our approach for generating Disambiguation Trees and for discovering infobox slots and fillers. We have evaluated the different steps in the framework individually. We plan to compare the individual steps with existing competing approaches and evaluate the integrated broader framework as a whole in the future.

References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., and Ives, Z. 2007. DBpedia: A nucleus for a web of open data. In ISWC '07: 11–15.

Bollacker, K., Cook, R., and Tufts, P. 2007. Freebase: A Shared Database of Structured General Human Knowledge. Proc. National Conference on Artificial Intelligence (Volume 2): 1962-1963.

Bizer, C. 2009. The Emerging Web of Linked Data. IEEE Intelligent Systems, v24, n5, pp. 87-92, Sep./Oct. 2009.

Cucerzan, S. 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. Proceedings of EMNLP-CoNLL 2007. pp. 708-716.

Finin, T. and Syed, Z. 2010. Creating and Exploiting a Web of Semantic Data, Proc. 2nd Int. Conf. on Agents and Artificial Intelligence, Valencia.

Finin, T., Syed, Z., Mayfield, J., McNamee, P. and Piatko, C. 2009. Using Wikitology for cross-document entity coreference resolution. In Proc. of the AAAI Spring Symposium on Learning by Reading and Learning to Read, AAAI Press.

Garera, N. and Yarowsky, D. (2009). Structural, transitive and latent models for biographic fact extraction, Proc. of EACL '09, pp. 300-308.

Hu, M., Lim, E.P., Sun, A., Lauw, H.W., and Vuong, B.Q. 2007. Measuring article quality in wikipedia: models and evaluation. In CIKM '07, New York, NY, USA.

Kazama, J. and Torisawa, K. 2007. Exploiting Wikipedia as external knowledge for named entity recognition. In Proc. of EMNLP-CoNLL'07: 698–707.

McNamee, P. and Dang, H. 2009. Overview of the TAC 2009 knowledge base population track. In Proc. of the 2009 Text Analysis Conference, NIST.

McNamee, P., Dredze, M., Gerber, A., Garera, N., Finin, T., Mayfield, J., Piatko, C., Rao, D., Yarowsky, D., Dreyer, M. 2009. HLTCOE Approaches to Knowledge Base Population at TAC 2009. In proc. of TAC '09.

Mihalcea, R. and Csomai, A. 2007. Wikify!: linking documents to encyclopedic knowledge. In Proc. of CIKM '07: 233–242.

Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. 1990. WordNet: An on-line lexical database. International Journal of Lexicography, 3:235–244.

Milne, D. and Witten, I.H. 2008. Learning to Link with Wikipedia. In Proc. of CIKM'08, pages 509-518.

Ponzetto, S.P. and Navigli, R. 2009. Large-scale taxonomy mapping for restructuring and integrating Wikipedia. In Proc. of IJCAI'09: 2083–2088.

Schonhofen, P. 2006. Identifying Document Topics using the Wikipedia Category Network. In Proc. of the 2006 IEEE/WIC/ACM international Conference on Web Intelligence Dec. 18-22, 2006.

Strassel, S., Przybocki, M., Peterson, K., Song, Z. and Maeda, K. 2008. Linguistic Resources and Evaluation Techniques for Evaluation of Cross-Document Automatic Content Extraction, Proceedings of the 6th Language Resources and Evaluation Conference, May 2008.

Suchanek, F.M., Kasneci, G., and Weikum, G. 2008. Yago: A large ontology from Wikipedia and WordNet. Web Semantics, 6(3):203–217.

Syed, Z., Finin, T. and Joshi, A. 2008. Wikipedia as an ontology for describing documents. In Proc: ICWSM'08. AAAI Press.

Syed, Z., Finin, T., Mulwad, V. and Joshi, A. 2010. Exploiting a Web of Semantic Data for Interpreting Tables, In Proc. of the Second Web Science Conference, April 2010.

Weld, D.S., Hoffmann, R., and Wu, F. 2008. Using Wikipedia to bootstrap open information extraction. SIGMOD Rec., 37(4):62–68.

Wu, F. and Weld, D.S. 2008. Automatically refining the Wikipedia infobox ontology. In WWW '08, pages 635–644.