A Novel Mining Technique for Linking Named Entities with Knowledge Base via Semantic Knowledge

Samata Peesapati [#1], Ch. Bindu Madhuri [*2]

[#1]M.Tech Scholar, [*2] Assistant.Professor
Department of Computer Science & Engineering,
University College Of Engineering,Vizianagaram ,JNTUK
Vizianagaram Dist,AP,India.

Abstract

A critical step in bridging the knowledge base with the huge corpus of semi-structured Web list data is to link the entity mentions that appear in the Web lists with the corresponding real world entities in the knowledge base, which we call list linking task. This task can facilitate many different tasks such as knowledge base population, entity search and table annotation. Named entity disambiguation is the task of disambiguating named entity mentions in natural language text and link them to their corresponding entries in the existing knowledge base. Such disambiguation can help enhance readability and add semantics to plain text. However, this task is challenging due to name ambiguity, textual inconsistency, and lack of world knowledge in the knowledge base. Several methods have been proposed to tackle this problem, but they are largely based on the co-occurrence statistics of terms between the text around the entity mention and the document associated with the entity. In this paper, we propose LINDEN, a novel framework to link named entities in text with a knowledge base unifying Wikipedia and Word-Net, by leveraging the rich semantic knowledge embedded in the Wikipedia and the taxonomy of the knowledge base. We extensively evaluate the performance of our proposed LINDEN over two public data sets and empirical results show that LINDEN significantly outperforms the state-of-the-art methods in terms of accuracy.

Keywords: Entity linking, Knowledge base, Fact integration, Semantic knowledge, Entity Disambiguation, List linking.

1. Introduction

The ability to identify the *named entities* (such as people and locations) has been established as an important task in several areas, including topic detection and tracking, machine translation, and information retrieval. Its goal is the identification of mentions of entities in text (also referred to as *surface forms* henceforth), and their labeling with one of several entity type labels. Note that an entity (such as George W. Bush, the current president of the U.S.) can be referred to by multiple surface forms (e.g., "George Bush" and "Bush") and a surface form (e.g., "Bush") can refer to multiple entities (e.g., two U.S. presidents, the football player Reggie Bush and the rock band called Bush)

Search engine has become the most convenient way for people to find their information on the Web, which is the world's largest encyclopedic source. Unfortunately, in response to the query for the facts or specific attributes about certain named entity, search engine always returns a flat, long list of Web pages containing the name of that entity. The users are then forced either to refine their queries by adding new keywords or to browse through every returned Web page which is quite time consuming. Therefore, the trend to advance the functionality of search engine to a more expressive semantic level has attracted a lot of attention in recent years. To achieve this goal, it is a vital step to construct a comprehensive machine-readable knowledge base about the world's entities, their semantic classes and their mutual relationships. Recently, many large scale publicly available

knowledge bases including DBpedia [1], YAGO [2, 3**]** and KOG [4, 5] have emerged.

The list linking task is of practical importance and can be used in various applications. For instance, 75% of the tables on the Web typically have a column that is the subject of the table, and the subject column contains the set of entities the table is about [6]. Linking this subject column with the knowledge base is significantly helpful for the task of table annotation [7] and recovering the semantics of tables **.** As another example, linking the Web lists or table columns with a knowledge base can enrich the existing knowledge base and impulse the trend to advance the traditional keyword-based search to the semantic entity-based search.

The emergence of large scale knowledge bases has spurred great interests in the entity linking task. Several methods [8, 9, 10] have been proposed to address this problem and they all aim to map the entity mention to its corresponding entity page in Wikipedia. Generally speaking, the essential step of entity linking is to define a similarity measure between the text around the entity mention and the document associated with the entity. Previous proposed methods [8, 9, 10] all use the bag of words model to measure the context similarity and consider this kind of similarity as an important feature to make the final decision. The bag of words model represents the context as a term vector consisting of the terms occurring in the window of text and their associated weights. Here, "terms "means words, phrases, named entities or Wikipedia concepts depending on the different methods.

Anyway, in the bag of words model, similarity is measured by the co-occurrence statistics of terms and cannot capture various semantic relations existing between concepts. The entity mention would be mapped to the corresponding entity in knowledge base only if the compared texts contain some identical contextual terms. However, by leveraging the semantic relation existing between concepts, the similarity can also be bridged by the semantically related concepts. For instance, we assume the knowledge base contains the following two entities which could be referred by the same name "Michael Jordan":

• Entity name : **Michael J. Jordan**

Description text : **American basketball player**

• Entity name : **Michael I. Jordan**

Description text : **Berkeley professor in AI**

When the entity mention appears in the text "Michael Jordan wins NBA champion.", we should map this occurrence of "Michael Jordan" to the American basketball player, because the concept"NBA"around the entity mention is highly semantically related to "American" and "Basketball" which are the concepts appearing in the description text associated with the entity "Michael J. Jordan". While in this situation, the bag of words model cannot work well.

In this paper, we propose LINDEN, a novel framework to link named entities in text with a knowledge base unifying Wikipedia and WordNet by leveraging the semantic knowledge derived from Wikipedia and the taxonomy of the knowledge base. It is assumed that the named entity recognition process has been completed, and we focus on the task of linking the detected named entity mention with the knowledge base. Specifically, we collect a dictionary about the surface forms of entities from four sources in Wikipedia (i.e., entity pages, redirect pages, disambiguation pages and hyperlinks in Wikipedia article), and record the count information for each target entity in the dictionary. Using this dictionary, we can generate a candidate entity list for each entity mention and try to include all the possible corresponding entities of that mention in the generated list. Furthermore, we leverage the count information to define the link probability for each candidate entity. Subsequently, we recognize all the Wikipedia concepts in the document where the entity mention appears.

Furthermore, LINDEN learns how to return NIL for the entity mention which has no matching entity in the knowledge base. To validate the effectiveness of LINDEN, we empirically evaluate it over two public data sets (i.e., Cucerzan's ground truth data [9] and the standard TAC2 data set The experimental results show that LINDEN greatly outperforms the previous methods in terms of accuracy. The main contributions of this paper are summarized as follows.

- We present LINDEN, a novel framework which leverages the rich semantic information derived from Wikipedia and the taxonomy of the knowledge base to deal with the entity linking task.
- We propose a novel method to measure the semantic similarity between Wikipedia concepts based on the taxonomy of the knowledge base.
- We extensively evaluate LINDEN for the entity linking task over two public data sets. The experimental results show that LINDEN can achieve significantly higher accuracy on both data sets compared with the state-of-the-art methods.

2. Related Work

The world knowledge used includes the known entities (most articles in Wikipedia are associated to an entity/concept), their entity class when available (Person, Location, Organization, and Miscellaneous), their known surface forms (terms that are used to mention the entities in text), contextual evidence (words or other entities that describe or co-occur with an entity), and category tags (which describe topics to which an entity belongs to).

Name ambiguity is very common on the Web and has raised serious problems in many different areas such as Web people search, question answering and knowledge base population. Before the emergence of large scale publicly available knowledge bases, named entity disambiguation is called coreference resolution and is regarded as a clustering task. Entity mentions of a particular name either within one document or across multiple documents are clustered together, and each resulting cluster represents one specific real world entity. This problem has been addressed by many researchers starting from Bagga and Baldwin **,** who used the bag of words model to represent the context of the entity mention and applied the agglomerative clustering technique based on the vector cosine similarity. Mann and Yarowsky [7] extended the work by adding a rich feature space of biographic facts. Pedersen et al. [11] employed the statistically significant bigrams to represent the context of a name observation. After that, several methods tried to capture the semantic relation between terms via constructing social networks to add the background knowledge for disambiguation. The work in [12] adopted the graph based framework to extend the similarity metric to disambiguate the entity mentions effectively. However, all these studies focus on clustering all mentions of an entity within a given corpus, which are insufficient for the entity linking task.

d	A document to be processed
M_0	All named entity mentions in d
$m \in M_0$	A named entity mention required to be linked
E	All entities in KB
$e \in E$	An entity label, here, the entity name in KB
E_m	The set of candidate entities for mention m
E_0	All candidate entities for all mentions in M_0
NIL	The label for the unlinkable mention
Γ_d	The set of context concepts in d
$F_m(e)$	The feature vector for entity $e \in E_m$
$\vec{w}$	Weight vector
$Score_m(e)$	Score of entity $e \in E_m$
τ	Threshold for returning NIL
$LP(e\|m)$	The *link probability* of entity e, given m
$SA(e)$	*Semantic associativity* of entity e with Γ_d
$SS(e)$	*Semantic similarity* of entity e with Γ_d
$GC(e)$	*Global coherence* of entity e in d

Table 1: Notations

The task of entity linking is similar to the lexical task of word sense disambiguation (WSD) in some aspects. The task of WSD aims to assign dictionary meanings to all instances of a predefined set of polysemous words in a corpora **.** For instance, it has to choose whether the word "tree" in some specific context refers to the meaning of plant or data structure in the field of computer science. Recently, people start to use Wikipedia as a resource for word sense disambiguation. Given an input document, these systems are able to automatically enrich the input text with links to Wikipedia pages [13]. However, this task is different from our entity linking task in several respects: firstly, these systems have to decide whether the detected terms or phrases are important enough in the document to be linked to Wikipedia due to considering the system users' experience, which raises the problem of tradeoff

between precision and recall. On the contrary, entity linking is the task to just map every detected entity mention in the text to the knowledge base to pursue high accuracy. Secondly, the named entity mentions like common person or place names have much higher average ambiguity compared with the keywords or concepts in the task of word sense disambiguation. Therefore, the entity linking task has much more challenges in comparison with the WSD task. Thirdly, the entity linking task has to encounter the problem that some entity mentions have no matching entities in the knowledge base. Consequently, it must learn how to predict NIL for the unlinkable mentions, while the word sense disambiguation task has no such problem.

3. The Linden Framework and Notations

In this section, we begin by describing the knowledge base and the task of list linking. Next, we introduce the generation of candidate mapping entities for each list item. In this paper, entity linking is defined as the task to map a textual named entity mention *m*, already recognized in the unstructured text, to the corresponding real world entity *e* in the knowledge base. If the matching entity *e* for entity mention *m* does not exist in the knowledge base, we should return NIL for *m*. The knowledge base we adopt in this work is YAGO [2, 3], an open-domain ontology combining Wikipedia and WordNet with high coverage and quality.

The reasons why we choose YAGO as the knowledge base are as follows. On one hand, YAGO has the vast amount of entities in the same order of magnitude as Wikipedia. On the other hand, it adopts the clean taxonomy of concepts from WordNet [14] which can be made fully use of by our LINDEN. Currently, YAGO contains over one million entities and five million facts about them. We process one document at a time, so we consider the entity mentions appearing in one document together. Given an input document *d*, $M0$ is the set of named entity mentions which need to be mapped in *d*. A named entity mention $m \in M0$ is a token sequence of a named entity that is potentially linked with an entity in the knowledge base, which has been detected beforehand. *E* is the set of all entities in the knowledge base, and an entity is expressed as the entity name in the knowledge base and denoted as *e*. Since some mentions' mapping entities do not exist in the knowledge base, we define this kind of mentions as unlinkable mentions and give NIL as a special label denoting "unlinkable". In this paper, we propose LINDEN, a framework to address this entity linking task with three modules as follows:

❖ Candidate Entity Generation

For each named entity mention $m \in M0$, we retrieve the set of candidate entities *Em* in this module. Using a dictionary collected from four sources in Wikipedia (i.e., entity pages, redirect pages, disambiguation pages and hyperlinks in Wikipedia article), we try to include all the possible candidate entities for each $m \in M0$ in *Em*. $E0$ is the set of all candidate entities for all mentions in $M0$.

❖ Named Entity Disambiguation

In most cases, the size of *Em* is larger than one, so we define a scoring measure for each $e \in Em$ and give a rank to *Em* to find which entity $e \in Em$ is the mostly likely link for *m*. We firstly recognize all the Wikipedia concepts Γd in the context of *d* and regard them as *context concepts* to represent the context of *d*. And then we define a rich set of features and generate a feature vector $Fm(e)$ for each $e \in Em$. The features used in LINDEN are mainly based on the *link probability LP* $(e|m)$, *semantic associativity SA* (e) of entity *e* with the context concepts in Γd derived from the Wikipedia link structure, *semantic similarity* $SS(e)$ of entity *e* with the context concepts in Γd measured from the taxonomy of YAGO, and *global coherence* $GC(e)$ of entity *e* with the other mapping entities associated with the mentions $\boldsymbol{m^1 \neq m \in M0}$**.** We also learn a weight vector $\vec{w}$ which gives different weights for each feature element in $Fm(e)$. Then we can calculate a score $\vec{w} \cdot Fm(e)$ for each $e \in Em$ and rank the candidates according to their $Scorem(e)$.

- **Unlinkable Mention Prediction**

  To deal with the problem of predicting unlinkable mentions, we learn a threshold τ in this module to validate whether the entity e_{top} which has the highest score in *Em* is the target entity for mention *m*. If $Scorem(e_{top})$ is smaller than the learned threshold τ, we return NIL for mention *m*.

Those three modules are introduced in the following sections in details and some notations used in this paper are summarized in Table 1.

4. Candidate Entity Generation

Given an entity mention $m \in M0$, we generate the set of candidate entities *Em* in this module. Intuitively, the candidates in *Em* should have the name of the surface form of *m*. To solve this problem, we need to build a dictionary that contains vast amount of information about the surface forms of entities, like name variations, abbreviations, confusable names, spelling variations, nicknames, etc. We take advantage of the huge amount of knowledge available in Wikipedia, a free online encyclopedia created through decentralized, collective efforts of thousands of users3. Wikipedia is the largest encyclopedia in the world and is also a very dynamic and quickly growing resource. English Wikipedia contains over 3,500,000 articles and new articles are added within days after their occurrence. The structure of Wikipedia provides a set of useful features for the construction of the dictionary we need, such as redirect pages, disambiguation pages and hyperlinks in Wikipedia article.

1) Entity page

Each entity page in Wikipedia describes a single entity, and generally, the title of each entity page is the most common name for that entity, e.g., the page title "IBM" for that giant American company headquartered in Armonk. Thus, we add the title of the entity page to the key *K*, and add the entity described in this page to **K.value.**

2) Redirect page

A redirect page exists for each alternative name which can be used to refer to an existing entity in Wikipedia. For example, the redirect page titled "HP" contains a pointer to the entity page titled "Hewlett-Packard". Henceforth, we add the title of the redirect page to the key *K*, and add the pointed entity to **K.value.**

3) Disambiguation page

When multiple entities in Wikipedia could be given the same name, a disambiguation page is created to separate them and contains a list of references to those entities. For instance, the disambiguation page for the name "Michael Jordan" lists eight associated entities having the same name of "Michael Jordan", including the famous NBA player and the Berkeley professor. For each disambiguation page, the title of this page is added to the key *K*, and the entities listed in this page are added to **K.value.**

4) Hyperlink in Wikipedia article

The article in Wikipedia often contains some hyperlinks each of which links to the page of the corresponding entity mentioned in this article. For example, in the entity page titled "Hewlett-Packard", there is a hyperlink pointing to the entity *William Reddington Hewlett* whose anchor text is "Bill Hewlett". Then we add the anchor text of the hyperlink to the key *K*, and add the pointed entity to *K.value*.

K (Mention form)	*K.value* (Mapping entity)
IBM	*IBM*
HP	*Hewlett-Packard*
Michael Jordan	*Michael Jordan* *Michael I. Jordan* *Michael Jordan (mycologist)* *Michael Jordan (footballer)* ...
Bill Hewlett	*William Reddington Hewlett*

Table 2: A part of the dictionary *D*

Using the four structures of Wikipedia described above, we construct the dictionary *D*. A part of the dictionary *D* is shown in Table 2.

5. Named Entity Disambiguation

In this section, we describe how to give a rank to Em when the size of Em generated in Section 4 is larger than one. Our guiding premise is that a document largely refers to coherent entities or concepts from one or a few related topics, and we exploit this "topical coherence" for named entity disambiguation. To achieve this goal, we firstly recognize all the Wikipedia concepts Γd in the document d, and by leveraging the rich semantic knowledge embedded in Wikipedia and YAGO, we construct a semantic network among the recognized Wikipedia concepts Γd and candidate entities E0, which will be described in Section 5.1.

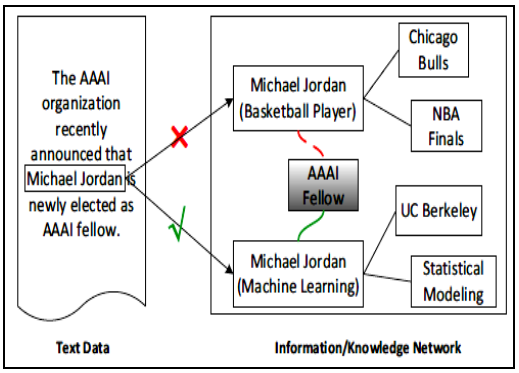

Figure 1: Named Entity Disambiguation Example

For example, as shown in Figure 1, if the extracted information "elected as AAAI fellow" is wrongly associated with the basketball player Michael Jordan, the network will lose the information that Michael Jordan (Machine Learning) is an AAAI fellow, as well as wrongly including Michael Jordan (Basketball Player) as a fellow of AAAI.

5.1 Semantic Network Construction

To construct the semantic network, we start by recognizing the Wikipedia concepts Γ*d* in the context of the document *d*, and regard them as context concepts to represent the context of *d*. For the general textual document, we utilize the open source toolkit Wikipedia-Miner to detect the Wikipedia concepts appearing in the context. The Wikipedia-Miner toolkit takes the general unstructured text as input and uses the machine learning approach to detect the Wikipedia concepts in the input document [15]. For instance, the entity mention of "Michael Jordan" occurs in a document containing such a sentence, "The Chicago Bulls' player Michael Jordan won his first NBA championship in 1991." For this sentence, we firstly remove the entity mention, and then utilize this Wikipedia-Miner toolkit to obtain four Wikipedia concepts, i.e., Chicago Bulls, National Basketball Association, NBA Finals and Chicago. Therefore, it can be seen that these detected Wikipedia concepts are highly semantically related to the NBA player Michael Jordan, and we can leverage this semantic information to link this entity mention "Michael Jordan" with the corresponding real world entity (i.e., the NBA player Michael Jordan) in the knowledge base effectively.

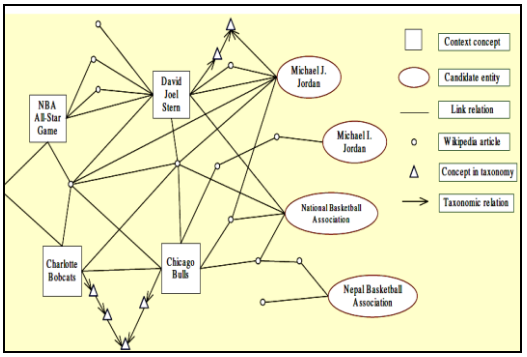

Figure 2: An example of the constructed semantic network

Figure 2 shows an example of the constructed semantic network. The four candidate entities in Figure 1 are generated from two entity mentions (i.e., "Michael Jordan" and "NBA"), and each of the entity mentions has two candidate entities respectively. From the constructed semantic network, we can see that the candidate entities "Michael J. Jordan" and "National Basketball Association" are more semantically related to the four context concepts compared with the other two candidate entities. Moreover, the semantic relations between "Michael J. Jordan" and "National

Basketball Association" also show the highly global topical coherence. Therefore, we can predict that "Michael J. Jordan" and "National Basketball Association" are the mapping entities for the entity mentions"Michael Jordan"and"NBA", respectively.

5.2 Semantic Associativity

Though the link relations among the context concepts Γ*d* and candidate entities *E*0 in Figure 2 express high semantic relations, this structure does not explicitly provide the exact value of the semantic relation's strength. In order to measure the strength of the link relation, we adopt the Wikipedia Link-based Measure (WLM) described in [16] to calculate the *semantic associativity* between Wikipedia concepts. Since all the context concepts Γ*d* and candidate entities *E*0 in our work are Wikipedia concepts, we can leverage this measure of WLM directly. The WLM which is modeled from the Normalized Google Distance [5] is based on Wikipedia's hyperlink structure. Given two Wikipedia concepts *e*1 and *e*2, we define the *semantic associativity* between them as follows:

$$SmtAss(e_1, e_2) = 1 - \frac{log(max(|E_1|, |E_2|)) - log(|E_1 \bigcap E_2|)}{log(|W|) - log(min(|E_1|, |E_2|))}$$

where *E*1 and *E*2 are the sets of Wikipedia concepts that link to *e*1 and *e*2 respectively, and *W* is the set of all concepts in Wikipedia.

5.3 Semantic Similarity

In this subsection, we propose a novel method to measure the *semantic similarity* between Wikipedia concepts based on the taxonomy of the knowledge base. According to the rules of constructing YAGO ontology in [3], each Wikipedia concept may have multiple super classes in the taxonomy. Given two Wikipedia concepts *e*1 and *e*2, we assume the sets of their super classes are Φ*e*1 and Φ*e*2, respectively. To measure the *semantic similarity* between Wikipedia concepts, we firstly define how to calculate the *semantic similarity*
between the sets of their super classes.

$$\varepsilon(C_1) = \arg\max_{C_2 \in \Phi_{e_2}} sim(C_1, C_2)$$

Where *sim*(*C*1, *C*2) is the *semantic similarity* between two classes *C*1 and *C*2, and $\varepsilon(C1)$ is the class in Φ*e*2 which maximizes the *semantic similarity* between these two classes.

5.4 Global Coherence

In this subsection, we exploit the global document-level topical coherence among entities which should be linked with by the mentions in M0. In this work, the global coherence GC(e) of entity e is measured as the average semantic associativity of entity e to the mapping entities of the other mentions m^1, where $m^1 \neq m \in M._0$ If em^1 is the mapping entity of mention m^1, then for entity e, the global coherence GC(e) is defined as

$$GC(e) = \frac{\Sigma_{m' \neq m \in M_0}(SmtAss(e_{m'}, e))}{|M_0| - 1}$$

6. Conclusion

In this paper, we propose LINDEN, a novel framework to link named entities in text with YAGO, a knowledge base unifying Wikipedia and WordNet. By leveraging the rich semantic knowledge derived from the Wikipedia and the taxonomy of YAGO, LINDEN can obtain great results on the entity linking task. A large number of experiments were conducted over two public data sets, i.e., the CZ data set and the TAC-KBP2009 data set. Empirical results show that LINDEN significantly outperforms the state-of-the-art methods in terms of accuracy. Moreover, all features adopted by LINDEN are quite effective for the entity linking task.

7. References

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of ISWC*, pages 11–15, 2007.

[2] F. Suchanek, G. Kasneci, and G. Weikum. Yago: A Large Ontology from Wikipedia and WordNet. *Journal of Web Semantics*, 6(3):203–217, 2008.

[3] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge unifying wordnet and wikipedia. In *Proceedings of WWW*, pages 697–706, 2007.

[4] F. Wu and D. S. Weld. Autonomously semantifying wikipedia. In *Proceedings of CIKM*, pages 41–50, 2007.

[5] F. Wu and D. S. Weld. Automatically refining the wikipedia infobox ontology. In *Proceedings of WWW*, pages 635–644, 2008.

[6] P. Venetis, A. Halevy, J. Madhavan, M. Pa¸sca, W. Shen, F. Wu, G. Miao, and C. Wu. Recovering semantics of tables on the web. *Proc. VLDB Endow.*, 4:528–538, June 2011.

[7] G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. *Proc. VLDB Endow.*, 3:1338–1347, September 2010.

[8] R. Bunescu and M. Pasca. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proceedings of EACL*, pages 9–16, 2006.
.
[9] S. Cucerzan. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of EMNLP-CoNLL*, pages 708–716, 2007.

[10] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin. Entity disambiguation for knowledge base population. In *Proceedings of COLING*, pages 277–285, 2010.

[11] T. Pedersen, A. Purandare, and A. Kulkarni. Name Discrimination by Clustering Similar Contexts. In *Proceedings of CICLing*, pages 226–237, 2005.

[12] E. Minkov, W. W. Cohen, and A. Y. Ng. Contextual search and name disambiguation in email using graphs. In *Proceedings of SIGIR*, pages 27–34, 2006.

[13] R. Mihalcea and A. Csomai. Wikify! Linking documents to encyclopedic knowledge. In *Proceedings of CIKM*, pages 233–242, 2007.

[14] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[15] D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceeding of CIKM*, pages 509–518,2008.

[16] D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceedings of WIKIAI*, 2008.

8. About the Authors

Samata Peesapati is currently pursuing her M.Tech in Computer Science & Engineering at University College of Engineering, Vizianagaram JNTUK. Her area of interests includes Security.

Mrs. Ch. Bindu Madhuri is currently working as a Assistant Professor in Department of IT, at University College of Engineering, Vizianagaram JNTUK. Her research interests include Data Mining, Web mining and Web usage mining.