Volume 8, No. 3, March – April 2017
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
REVIEW ARTICLE
Available Online at www.ijarcs.info

Techniques For Efficient Short Text Understanding: A Survey on Related Literature

Geena Jojy
M.Tech Student (Information Systems)
Department of Computer Science & Engineering
FISAT, Mookkannoor, Kerala, India

Reshmi R
Assistant Professor
Department of Computer Science & Engineering
FISAT, Mookkannoor, Kerala, India

Abstract: The trend of social media and various online applications has rapidly increased over the past few years. These computer-mediated communications has resulted in the generation of large amount of short texts. A short text refers to the text with limited contextual information. Lots of interest lies in analyzing and conceptualizing short text for understanding user intents from search queries or mining social media messages. Consequently, the task of understanding short text is crucial to many online applications. But it is not ease to handle enormous volume of short texts, since they are relatively more ambiguous and noisy than normal data. The short texts do not follow the syntax of natural language. Thus, point out the necessity for an efficient text understanding technique. The task of short text understanding or conceptualization can be divided into three, as text segmentation, type detection, and concept labeling. In text segmentation, initially the input text is processed and removes all the stop words if any. Then it is divided into a sequence of terms. POS tagging decide the lexical types (i.e. POS tags) of terms in a text. Type detection is incorporated into the framework for short text understanding and it help to conduct disambiguation based on various types of contextual information that present in the text. Finally, concept labeling is performed to discover the hidden semantics from a natural language text. The conceptualization can benefit from various online applications such as automatic question-answering, recommendation systems, online advertising, and search engines. All these applications requires an information extraction phase in which the prior step is to extract the concepts from the input text. Now-a-days conceptualization is used to develop machine learning techniques for information extraction. Hence the task of conceptualization or short text understanding plays a vital role in the area of machine learning, which is an active area of research. In this paper, the current techniques used for text segmentation, type detection, and concept labeling are reviewed.

Keywords: Short text understanding; conceptualization; semantic labeling; text segmentation; part-of-speech tagging

I. INTRODUCTION

Huge explosion of information urge the need for machines that better understand natural language texts. The short text refers to those groups of words or phrases with limited context, that are generated via search queries, twitter messages, ad keywords, captions, document titles etc. So, a better understanding of a short text disinters the hidden semantics from texts. Also lot of interests lies in analyzing and conceptualizing short text for understanding user intents from search queries or mining social media messages for business insights. But understanding short text is a challenging task for machine intelligence meanwhile a very relevant concept on handling massive text data.

As stated in Psychologist Gregory Murphy's highly acclaimed book, "Concepts are the glue that holds our mental world together" [1]. Therefore, conceptualization illustrates a short text to a set of concepts, concept space, as a mechanism of understanding short texts. It is doubtless to say, the ability to conceptualize is a defining characteristics of humanity. Humans are capable of forming rich models of the world and make strong generalizations from input data that is noisy, ambiguous and sparse. The problem is can machines do it? Even though short texts understanding will bring tremendous values, the task still abounds with lots of challenges.

An important challenge that would be faced while dealt with short texts is that they do not always follow the syntax of a written language. Also short texts usually do not have sufficient content to support statistical models. It may usually be informal and error-prone i.e., short texts are noisy and may have ambiguous types. A typical strategy for short text understanding mainly consists of three steps according to [2]:

1) Text segmentation: Splits a short text into a collection of terms (i.e. words and phrases) contained in a vocabulary (e.g. "book SeaGate hotel Goa" is segmented as {book SeaGate hotel Goa}).

2) Type detection: Determines the type of terms and recognize instances (e.g. both "SeaGate" and "Goa" are recognized as instances (e), while "book" is recognized as verb (v) and "hotel" a concept(c)).

3) Concept labeling: Derive the concept of each instance (e.g. "SeaGate" and "Goa" refer to the concept *theme park* and *state* respectively.

Overall, three concepts are detected from the short text "book SeaGate hotel Goa" using this strategy, namely theme park, hotel, and state as shown in Fig. 1.

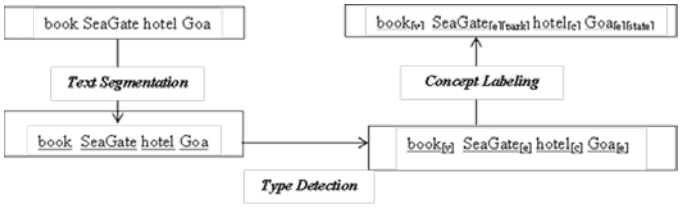

Figure 1. An example of short text understanding

The rest of this paper is organized as follows: section II describes the different approaches existing for text segmentation, type detection, and semantic labeling; related works in the literature of text processing are mentioned in section III; followed by a brief conclusion in section IV.

II. BACKGROUND STUDY

This section specifies different methods used for text understanding. The first step for conceptualization or short text understanding is text segmentation. Text segmentation is considered as dividing a text into sequence of terms. Existing

approaches for text segmentation can be classified into two categories: statistical approach and vocabulary-based approach. In **statistical approaches**, the frequencies of words occurring as neighbors in a training corpus are calculated. When the frequency exceeds a predefined threshold, the corresponding neighboring words can be treated as a term. While in **vocabulary-based approach**, terms are extracted in a streaming manner by checking for existence or frequency of a term in a predefined vocabulary.

Text segmentation divides the input text into sequence of terms, which is further assigned tags in POS tagging phase. POS tagging is the process of assigning a part-of-speech or lexical types of words in a sentence according to the context and is used in wide range of applications that include information extraction, word sense disambiguation etc. Process of POS Tagging: read the input sentence. Then tokenize the sentence into words. After tokenization, suffix and prefix analysis are also used for correctly tag each word of sentence. Then use one of the tagging methods to tag each word of sentence of corpus as noun, verb, conjunction, number tag etc. The output is tagged sentence. Mainstream POS tagging algorithm fall into two categories: rule-based approach and statistical approach. In **rule-based approach**, POS tags are assigned to unknown or ambiguous words based on a large number of handcrafted or automatically learned linguistic rules. While in the latter, a **statistical model** is automatically build from a corpora and labeling untagged text based on those learned statistical information. Both rule-based and statistical approaches rely on the hypothesis that texts are correctly structured i.e. text should satisfy tagging rules or sequential relations between consecutive tags. Statistical POS taggers avoid the cost of constructing tagging rules.

Semantics is a field of Natural Language Processing (NLP) concerned with extracting the meaning from a sentence. Semantic Labeling discovers the hidden semantics from a text. Existing works related to semantic labeling can be categorized into three categories, based on the representation of semantics, which includes named entity recognition (NER), topic modeling and entity linking. **NER** locates named entities in a text and a linguistic grammar-based technique as well as a statistical model classifies them into predefined categories (e.g., persons, organizations, quantities, locations and percentages etc.). Based on the observable statistical relations between texts and words, **topic models** recognize "latent topics" that are represented as probabilistic distributions on words. While **entity linking** uses existing knowledgebase and focus on retrieving "explicit topics" which are expressed as probabilistic distributions on the entire knowledgebase.

III. RELATED WORKS

In this section, related works are discussed in mainly three aspects for short text conceptualization: text segmentation, POS (Part-Of-Speech) tagging and semantic labeling.

A. Text segmentation

1) Statistical approach: A statistical model for domain-independent text segmentation is proposed in [3]. The model described in this work finds the maximum probability segmentation of a given text. Since it estimates probability from given text, the method does not require training data. The text segmentation algorithm works as follows: it selects an optimum segmentation in terms of probability defined by a statistical model. Given a text of n words, $W = w_1,w_2,\ldots,w_n$, then the W can be segmented into m segments, $S_1,S_2,\ldots,S_m$. The probability of the segmentation, S and the most likely segmentation, $\hat{S}$ are calculated. Then the minimum cost segmentation or maximum probability segmentation can be trace out by finding a minimum cost path in a graph. Since this model does not require training data to estimate probabilities, it is also applicable to domain-independent texts. The method is more accurate than or atleast as accurate as a state-of-the-art text segmentation system.

An unsupervised query segmentation scheme is introduced that uses query logs as the only resource and can effectively capture the structural units in queries [4]. Here a statistical model based on Hoeffding's is applied to mine significant word n-grams from queries and subsequently use them for segmenting the queries. This technique can detect rare units missed by PMI baseline. Queries are neither bag-of-words nor grammatically correct language phrases or sentences but are considered as bag-of-units. The method described is as follows: Given a large collection of search queries. Consider an n-gram $M = (w_1,w_2,\ldots,w_n)$ where w_j's denote the words constituting M. Let $\{q_1,q_2,\ldots,q_k\}$ be the subset of queries in the database that contain all the words of M, though not necessarily occurring together as an n-gram. And the premise is that search queries can be viewed as bags of Multi-Word Expressions (MWE's), i.e. any permutation of the MWEs constituting a particular search query will effectively represent the same query.

Let focus on M, a candidate MWE, and models the number of times the words of M appear in the k queries. Use Hoeffding's Inequality to obtain an upper bound δ on the probability of $[X \geq N]$ where, N denotes the observed value of X in the data as mentioned in [4]. After obtaining δ for each n-gram M, define $(-\log \delta)$ as the MWE score for M. If δ is small, indicates a greater chance of M being an MWE led the surprise factor to be higher, and vice versa. Now have a list of significant n-grams and their associated MWE scores. This list is used to perform unsupervised query segmentation as follows: First compute a final score for each of the possible segmentation by summing the MWE scores of individual segments. Then pick the segmentation that yields the highest segmentation score. Here use a dynamic programming approach to search over all possible segmentations. The result shows that the scheme proposed here performs better than a baseline method that uses PMI. But on close examination of the segmentation results, found that many segments discovered by this scheme did not match with human annotations because segmentation done by human is largely influenced by natural language grammar.

Reference [5] proposes a novel way for modeling topics in short texts, referred as biterm topic model (BTM), has become an important task for many content analysis applications. Specifically, in BTM the topics by directly modeling the generation of word co-occurrence patterns (i.e. biterms) in the whole corpus are learned. The advantages of BTM are: BTM uses the aggregated patterns in the whole corpus for learning topics to solve the problem of sparse word co-occurrence patterns at document-level; and BTM explicitly models the word co-occurrence patterns to enhance the topic learning. The idea of BTM is to learn topics over short texts based on the aggregated biterms in the whole corpus to tackle the sparsity problem in single document. The probability that a biterm drawn from a specific topic is further captured by the chances that both words in the biterm are drawn from the topic.

2) Vocabulary-based approach: Reference [6] introduce a probabilistic knoledge base which is as rich as human mental world in terms of the concept it contains. A Bayesian inference mechanism is developed to conceptualize words and short texts. Statistical approaches such as topic models treat text as a bag of words in vector space, and discover the latent topics that are considered as set of words from given text. But finding latent topics is not commensurate with understanding the text. The mining results often have low interpretability since the machines ignore the semantics of the text largely. To enable machines to perform human-like conceptualization, this work proposes a probabilistic framework, which includes a knowledge base named Probase and certain inference techniques on top of the knowledgebase.

Here a method to infer concepts from a set of instances, or a set of attributes is illustrated. The problem is to identify candidate concepts ranked by their likelihood when observe a set of instances, or a set of attributes, or a set of terms of unknown types. Naïve bayes model is used to estimate the probability of concepts. And the concept with the largest posterior probability is ranked as the most possible concept to describe the observed instances. Same is used for conceptualizing attributes and concepts. The inference of relationships between attributes and a concept should be intermediated through the instances of the concept as well. Therefore, Bayes chain rule is applied to derive the likelihood of concepts. Compared with traditional latent semantic analysis and topic modeling such as Latent Dirichlet Allocation (LDA), explicit semantic analysis has the advantage of providing semantics that are interpretable by human beings. Also the use of knowledgebase, Probase, which is rich with millions of concepts and instances has improved accuracy a lot in clustering based applications.

B. Pos tagging

1) Rule based approach: A simple rule-based part-of-speech tagger is a tagger that works by automatically recognizing and remedying its weakness, thereby gradually improving its performance [7]. The tagger initially tags by assigning each word its most probable tag, estimated by examining a large tagged corpus, without regard to context. The initial tagger has two procedures built in to improve performance – one procedure is provided with information that words that are not in the training corpus and are capitalized tend to be proper nouns. The other procedure works for tagging words not seen in the training corpus by giving such words, the tag most common for words ending in the same three letters. Both the information could be acquired automatically from the training corpus. Once the initial tagger is trained, it is used to tag the patch corpus. The tagger then acquires patch templates to improve its performance. These patches are applied to remedy the mistagging of a word as tag_a when it should be tagged as tag_b. The patch which results in the greatest improvement is added to the list of patch corpus and further it can be used to tag new text to decrease the error rate. The authors claim to use a very simple algorithm with an error rate of about 7.9% when trained on 90% of the tagged Brown[1] corpus.

2) Statistical approach: An implementation of a part-of-speech tagger based on a hidden Markov model is presented in [8]. Two types of training (i.e. parameter estimation) have been used with this model. The first make use of a tagged training corpus. The second method of training does not need tagged training corpus. A lexicon and a suitably large sample of ordinary text are the only resoures required for this model hence taggers can be built with minimal effort even for other languages. The hidden Markov modeling component of tagger is implemented as an independent module. HMM is a process that generates sequence of symbols $S = S_1,S_2,\ldots,S_T$, $S_i \in W$ and described by a set of N states, a matrix of transition probabilities $A = \{a_{ij}\}$ where $1 \leq i, j \leq N$ and a vector of initial probabilities $\prod = \Pi_i$, $1 \leq i \leq N$. Hidden Markov Modeling allows computig the most probable sequence of state transitions, provided the parameters A, and $\prod$, and hence the mostly likely sequence of lexical tags, corresponding to sequence of ambiguity classes. N can identified with the number of possible tags, and W with the set of all ambiguity classes.

The POS tagger described here is implemented as an analysis module so the tagger generate terms from the text. In this scenario, a term is a word stem marked with part of speech. After entering the analysis sub-system, the first processing module encountered by the text is the tokenizer, whose duty is to convert text (a sequence of characters) into a sequence of tokens. The tokenizer subsequently passes tokens to the lexicon. Here tokens are converted into a set of stems, each annotated with a POS tag. The set of tags discovers an ambiguity class. The responsibility of the lexicon also includes the identification of these classes. Thus the lexicon delivers a set of stems paired with tags, and an ambiguity class. These long sequences of ambiguity classes are fed as input to the training module. It uses the Baum-Welch algorithm to produce a trained HMM, an input to the tagging module. Sequences of ambiguity classes between sentence boundaries are buffered by the tagging module and are disambiguated by computing the maximal path through the HMM with the Viterbi algorithm. The resulting sequence of tags is used to select the appropriate stems. Since sentence boundaries are unambiguous, operating at sentence granularity provides fast throughput without loss of accuracy.

Reference [9] introduced a tagging algorithm for English sentences based on Hidden Markov Model and Viterbi Algorithm. In traditional part-of-speech taggers, the calculation requires $(2T+1) * N^T+1$ multiplications if used the direct computation. After enhancing the method of calculating, get the optimal tags sequence by just $2N^2T$ multiplications, where T denotes the number of tags in a tag set. The Viterbi algorithm has three steps: (i) initialization, (ii) recursion, and (iii) termination. Here compute two functions $\gamma_i(j)$, which gives us the probability of being in state j tag at word i, and j, which gives the most likely state (or tag) at word i given that we are in state j (=tag j) at word i, and the function $\psi_{i+1}(j)$, which gives the most likely state (or tag) at word i given that

[1] The Brown Corpus contains about 1.1 million words from a variety of genres of written English. There are 192 tags in the tag set, 96 of which occur more than one hundred times in the corpus.

we are in state j at a word i+1. The initialization step is to assign probability 1.0 to the tag HASH #. Each sentence starts with a HASH and also ends with a HASH. i.e., assume that sentences are delimited by HASHs. Then, an algorithm is implemented to tag the sentences that are chosen randomly. After enhancing the method of calculating, get the optimal tags sequence by just $2N^2T$ multiplications. This method avoids the cost of constructing tagging rules, but only considers lexical features and ignores word semantics. Also calculate the ratios of each word tagging by hand.

C. Semantic labeling

1) Named entity recognition: Named entity recognition using statistical model, Conditional Random Field (CRF) is proposed in [10]. Conditional Random Fields (CRFs) are considered as undirected graphical models, its special case correspond to conditionally-trained finite state machines. They have efficient procedures for non-greedy, complete finite-state inference and training.

A named entity (NE) recognition (NER) system was built to recognize and classify names, times and numerical quantities using a Hidden Markov Model (HMM) and an HMM-based chunk tagger [11]. NER system uses two kinds of evidences to solve the problem regarding ambiguity and robustness – first is the internal evidence found within the word and/or word string itself while the second is the external evidence gathered from its context. Through the HMM, the system is able to apply and integrate four types of internal and external evidences: 1) internal gazetteer feature; 2) simple deterministic internal feature of the words, such as capitalization; 3) internal semantic feature of important triggers; 4) external macro context feature. Given a token sequence $G_1^n = g_1 g_2 \dots g_n$, the goal of NER is to find a stochastic optimal tag sequence $T_1^n = t_1 t_2 \dots t_n$ that maximizes log $P(T_1^n | G_1^n)$. It uses the mutual information between T_1^n and G_1^n. Here HMM is based on mutual information assumption instead of the conditional probability independence assumption. The author claims an F-measure of 96.6% on evaluating on system MUC-6. The performance is significantly better than reported by any other machine-learning system. Also, the performance is even consistently better than those based on handcrafted rules.

2) Topic models: A generative model, called author-topic model, as for documents that extends Latent Dirichlet Allocation (LDA) to include authorship information is specified in [12]. It describes a model for document collections, the author-topic model, which simultaneously models the content of documents and the interests of authors. Each document is represented with a mixture of topics as in state-of-the-art approach like LDA. Finally obtain the set of topics that appear in a corpus and their relevance to different documents.

For modeling documents with topics, the generation of document collections is modeled as a three step process. Initially, a distribution over topics is sampled from a Dirichlet distribution for each document. Then, a single topic is chosen according to this distribution for each word in the document. Ultimately, each word is sampled from a multinomial distribution over words specific to the sampled topic. This generative process correlates to the hierarchical Bayesian model. In this model, for each word, z denotes the topic responsible for generating that word, drawn from the document distribution Θ and w is the word itself drawn from the topic distribution ϕ corresponding to z. A variety of algorithms have been used to estimate these parameters, here use Gibbs sampling as it provides a simple method for obtaining parameter estimates under Dirichlet priors (α and β) and allows combination of estimates from several local maxima of the posterior distribution. For modeling authors with words, an author is chosen uniformly at random for each word in the document and a word is chosen from a probability distribution over words that are specific to that author. The author-topic model draws upon the strengths of two models defined above, using a topic-based representation to model both the content of documents and the interests on authors. The LDA model can adapt its distribution over topics to the content of individual documents even better as more words are observed. When compared to LDA topic model, the author-topic model shown to have more focused priors when relatively little is known about a new document.

3) Entity linking: The increased availability of large-scale, rich semantic knowledge source provides new opportunities to exploit these knowledge sources at the best to develop better algorithms for solving named entity disambiguation. The problem is that these knowledge sources possess semantic knowledge in complex structures, such as graphs and networks. Reference [13] proposed Structural Semantic Relatedness (SSR), a knowledge-based method which can solve named entity disambiguation by capturing and leveraging the structural semantic knowledge in multiple sources. A reliable semantic relatedness measure between concepts (in this paper uses WordNet and Wikipedia concepts) act as the key point in this method. Structural Semantic Relatedness (SSR) capture both the implicit semantic knowledge embedded in graphs and networks and the explicit semantic relations between concepts. Initially the semantic relations between two concepts are extracted from a variety of knowledge sources and represent them using a graph-based model, called semantic-graph. Then based on the principle that "if a concept is semantically related to the neighboring concepts then those concepts are semantically related to each other", construct SSR measure. The experimental results proved that SSR method can significantly outperform the traditional methods.

Another system was introduced that uses Wikipedia as a resource for automatic keyword extraction and word sense disambiguation [14]. The system identify the important concepts in a text (keyword extraction), and these concepts are linked to the corresponding Wikipedia pages (word sense disambiguation), when a document is given as input. When a text or hypertext document is given, "text wikification" task is performed initially that automatically extract important words and phrases in the document, and identify for each such keyword the appropriate link to a Wikipedia article. In order to overcome the problem of link ambiguity, the hypertexts are pre-processed by separating the HTML tags and the body text. The clean text is then passed to keyword extraction module, which implements an unsupervised keyword extraction algorithm that works in two steps - candidate extraction, and keyword ranking. The input document is parsed in candidate extraction step and extracts all possible n-grams that are also present in the controlled vocabulary. A numerical value is

assigned to each candidate, reviewing the likelihood that a given candidate is a valuable key phrase.

On experimental evaluations on three different ranking methods, the results for the traditional measures of *tf.idf* and χ^2 are very close to each other, while the keyphrase measure produces significant higher scores. After keyword extraction, word sense disambiguation is performed. Two different disambiguation algorithms are mentioned in this work – first one is knowledge-based approach and the other one is data-driven method. The most probable meaning for a word in a given context, which is a measure of contextual relatedness between the dictionary definitions of the ambiguous word, and the context where the ambiguous word occurs are identified in the Knowledge-based approach. In the latter method both local and topical features are integrated into a machine learning classifier and extract a training feature vector for each of its occurrences inside a Wikipedia link. Finally a voting scheme is used to filter out the incorrect predictions by seeking agreement between two methods. Evaluations of the system showed that the automatic annotations are reliable and hardly distinguishable from manual annotations.

A crucial step in bridging between unstructured Web text and semi structured search and mining applications is to identify "spots" or textual references to named entities and annotate the spots with unambiguous entity IDs (called "labels") from a catalog. Multiple systems have been proposed to link spots on Web pages to entities in Wikipedia, in which most of them are focus on local compatibility between the text around the spot and textual metadata associated with the entity. Reference [15] described a general collective disambiguation approach for annotating unstructured text (Web) with entity IDs from an entity catalog (Wikipedia). The new models and algorithms described in this work provide a high-recall open-domain annotation for indexing and mining tasks on comparing with the prior works, which is biased toward specific entity types like persons and places. The main contribution is a formulation that captures a tradeoff between local spot-to-label compatibility and a global document-level topical coherence between entity labels. Inference in this model is intractable in theory, but shown that LP relaxations often give optimal integral solutions or achieve close to the optimal objective. The work also specifies a simple local hill-climbing algorithm that is comparable in speed and quality to LP relaxation. Experimental results showed that both the algorithms are significantly better than prior works on annotation algorithms.

IV. CONCLUSION

For many text mining applications like classification and clustering the task of understanding short text is considered as an underlying task or an online task. It is known that these applications need to handle millions of short texts at a time, signifies the importance of an efficient text conceptualization or text understanding task. A short text understanding can be more specifically divided into three steps, as text segmentation, type detection and concept labeling. Since the efficiency of short text understanding is extremely critical, each of these steps is required to be more precise. This emphasizes the importance of the survey, in which state-of-the-art techniques for above mentioned steps are discussed. After the survey on related literature, it can be summarized as follows: (1) the drawback of existing methods for text segmentation is that they consider only surface features while ignore the semantic coherence within segmentation. (2) Considering type detection or POS tagging, rule based approach is very complex and also time consuming. Hence statistical methods like well-known Markov Model overwhelm rule based methods. (3) Regardless of the high accuracy achieved by existing works on semantic labeling, there are still some limitations or challenges abound.

In order to cope with the challenges and limitations in short text understanding, it requires three types of knowledge: (1) a comprehensive dictionary and vocabulary; (2) mappings between instance and concepts; (3) semantic relatedness or coherence between terms. Thus, survey points towards the need for a generalized framework that exploits the context semantics, so that better accuracy can be achieved while conducting the text understanding.

V. REFERENCES

[1] G. L. Murphy, "The big book of concepts," MIT press, 2004.

[2] W.Hua, Z. Wang, H. Wang, K.Zheng, and X. Zhou, "Understanding short texts by harvesting and analyzing semantic knowledge," IEEE transactions on Knowledge and data Engineering, Vol.29. No.3, March 2017.

[3] M. Utiyama and H. Isahara, "A statistical model for domain-independent text segmentation," in Proc. Of the 39th Annual meeting on Association for Computational Linguistics, ser ACL'01, Stroudsburg, PA, USA, 2001. Pp. 499-506.

[4] N. Mishra, R. Saha Roy, N. Gaguly, S. Laxman, and M. Choudhury, "Unsupervised query segmentation using only query logs," in Proc. Of the 20th International Conference Companion on World Wide Web, ser.WWW'11, 2011, pp.91-92.

[5] X. Yan, J. Guo, and X.Cheng, "A biterm topic model for short texts," International World Wide Web Conference Committee(IW3C2), ACM, 978-1-4503-2035.

[6] Y. Song, H.Wang, Z.Wang, H.Li, and W. Chen, "Short text conceptualization using a probabilistic knowledgebase," in Proc. Of the 22nd International Joint conference on Artificial Intelligence – Volume three, ser.IJCAI'11, 2011, pp. 2330-2336.

[7] E. Brill, "A simple rule-based part of speech tagger," in Proc. Of the workshop on Speech and Natural Language, ser. HLT'91, stroudsburg, PA, USA, 1992, pp. 112-116.

[8] D. Cutting, J. Kupiec, J. Pedersen, and L. P. Sibun, "A practical part-of-speech tagger," in Proc. of the 3rd conference on Applied natural language processing, ser. ANLC'92, Stroudsburg, PA, USA, 1992, pp.133-140.

[9] H. schutze and Y. Singer, "Part-of-speech tagging using a variable memory markov model," in Proc. of the 32nd annual meeting on Association for Computational Linguistics, ser.ACL'94, Stroudsburg, PA, USA, 1994, pp.181-187.

[10] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhaced lexicons," in Proc. of the 7th conference on Natural language learning at HLT-NAACL 2003 – Volume 4, ser.CONLL'03, Stroudsburg, PA, USA, 2003, pp. 188-191.

[10] G.Zhou and J. Su, "Named entity recognition using hmm-based chunk tagger," in Proc. of the 40th Annual meeting on Association for Computational Linguistics, ser. ACL'02, Stroudsburg, PA, USA, 2002, pp. 473-480.

[11] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in Proc. of the 20th Conference on Uncertainity in Artificial Intelligence, ser.UAI'04, Arlington, Virgina, United States, 2004, pp. 487-494.

[12] X. Han and J. Zhao, "Structural semantic relatedness: A knowledge-based method to named entity disambiguation," in Proc. of the 48th Annual Meeting of the Association for Computational Linguistics, ser. ACL'10, Stroudsburg, PA, USA, 2010, pp. 50-59.

[13] R. Mihalcea and A.csomai, "Wikify! Linking documents to encyclopedic knowledge," in Proc. of the 16th ACM conference on information and knowledge management," in ser. CIKM'07, New york, NY, USA, 2007 , pp. 233-242.

[14] S. Kulkarni, A. Singh, G. ramakrishnan, and S. Chakrabarti, "Collective annotation of wikipedia entities in web text," in Proc. of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, ser. KDD'09, New York, NY, USA, 2009, pp.457-466.