KNN TFIDF Based Named Entity Recognition

[1]B.Upendraa, [2]Dr. A. Sudheer Babu

[1]PG Scholar, [2]Professor
Department of CSE,
Prasad V Potluri Siddartha Institute of Technology, Vijayawada, AP, India

***ABSTRACT*---The increasing volume of generated crime information readily available on the web makes the process of retrieving and analyzing and use of the valuable information in such texts manually a very difficult task. This work is focus on designing models for extracting specific information from the Web. Thus, this paper proposes an ensemble framework for named entity recognition task. The main aim is to efficiently integrating feature sets and classification algorithms to synthesize a more accurate classification procedure. First, three well-known text classification algorithms, namely K-Means and K-Nearest Neighbor classifiers, are employed as base-classifiers for each of the feature sets. Second, weighted voting ensemble method is used to combine theses three classifiers. Experimental results demonstrate that using ensemble model is an effective way to combine different feature sets and classification algorithms for better classification performance.**

***KEYWORDS*--- NER domain, Classifiers, Combination method, Machine Learning**

I. INTRODUCTION

The task of Named Entity Recognition (NER) allows identifying proper names as well as temporal and numeric expressions, in an open-domain text. NER systems proved to be very important for many tasks in Natural Language Processing (NLP) such as Information Retrieval and Question Answering tasks. Existing NER systems have been constructed using mainly knowledge based or linguistic, and machine learning approach. The Knowledge based or linguistic approach is basically a rule-based approach which uses a set of handcrafted rules are designed and defined by human experts ,especially linguists. This model considers a set of patterns consisting of grammatical, syntactic, linguistic and orthographic features in combination with dictionaries. Machine learning approaches inherently supports rule-based systems or use sequence labeling algorithms to collect knowledge from a collection of training examples.
The semi-supervised is relatively recent technique which is basically a bootstrapping approach and involves a small degree of supervision in form of a set of seeds for knowledge acquisition. Unsupervised learning technique is a clustering technique based on the similarity of context, lexical patterns and other relevant features collected from lexical resources.
Named entity recognition is used in many applications throughout several domains and fields. Named Entity Recognition system is an essential component of complex information extraction system. A named entity tagger serves as a preprocessing step of machine translation system.

Problem definition

In the expression named entity, the word named restricts the task to those entities for which one or many rigid designators, as defined by Kripke, stands for the referent. For instance, the automotive company created by Henry Ford in 1903 is referred to as Ford or Ford Motor Company. Rigid designators include proper names as well as terms for certain biological species and substances.
Full named-entity recognition is often broken down, conceptually and possibly also in implementations, as two distinct problems: detection of names and classification of the names by the type of entity they refer to (e.g. person, organization, location and other). The first phase is typically simplified to a segmentation problem: names are defined to be contiguous spans of tokens, with no nesting, so that "Bank of America" is a single name, disregarding the fact that inside this name, the substring "America" is itself a name. This segmentation problem is formally similar to chunking.
Temporal expressions and some numerical expressions (i.e., money, percentages, etc.) may also be considered as named entities in the context of the NER task. While some instances of these types are good examples of rigid designators (e.g., the year 2001) there are also many invalid ones (e.g., I take my vacations in “June”). In the first case, the year 2001 refers to the 2001st year of the Gregorian calendar. In the second case, the month June may refer to the month of an undefined year (past June, next June, June 2020, etc.). It is arguable that the named entity definition is loosened in such cases for practical reasons. The definition of the term named entity is therefore not strict and often has to be explained in the context in which it is used.

II. Literature survey

Linking Named Entities to Any Database

This paper introduces a new task, called Open-Database Named-Entity Disambiguation (Open-DB NED), in which a system must be able to resolve named entities to symbols in an arbitrary database, without requiring labeled data for each new database. Existing techniques for disambiguating named entities in text mostly focus on Wikipedia as a target catalog of entities. The main

aim of open-DB NED is to resolve an entity to Wikipedia or to any relational database that meets mild conditions about the format of the data.
They introduce two techniques for Open-DB NED

i) One based on distant supervision
ii) The other based on domain adaptation.

The first strategy, a distant supervision approach, uses the relational information in a given database and a large corpus of unlabeled text to learn a database-specific model.
The second strategy, a domain adaptation approach, assumes a single source database that has accompanying labeled data.

Large-Scale Named Entity Disambiguation Based on Wikipedia Data

This paper presents a large-scale system for the recognition and semantic disambiguation of named entities based on information extracted from a large encyclopedic collection. It describes in detail the disambiguation paradigm employed and the information extraction process from Wikipedia.
In this paper the author describes how information extraction from Wikipedia and how the surface forms are mapped with entities.
The system described in this paper has been fully implemented as a Web browser, which can analyze any Web page or client text document. The application on a large scale of such an entity extraction and disambiguation system could result in a move from the current space of words to a space of concepts, which enables several paradigm shifts and opens new research directions, which we are currently investigating, from entity-based indexing and searching of document collections to personalized views of the Web through entity based user bookmarks.

Exploring Entity Relation for Name Entity Disambiguation

Named entity disambiguation is the task of linking an entity mention in a text to the correct real-world referent predefined in a knowledge base, and is a crucial subtask in many areas like information retrieval or topic detection and tracking. Named entity disambiguation is challenging because entity mentions can be ambiguous and an entity can be referenced by different surface forms.
In this paper, they evaluate a range of novel disambiguation features that exploit the relations between NEs identified in a document and in the KB. The main goal is to explore the usefulness of Wikipedia's link structure as source of relations between entities. They propose a method for candidate selection that is based on an inverted index of surface forms and entities. Instead of a bag-of-words approach they use co-occurring NEs in text for describing an ambiguous surface form.
In future work we plan to explore multilingual data for NED. Since non-English versions of Wikipedia often are less extensive than the English version we find it promising to combine Wikipedia versions of different languages and to use them as a source for multilingual NED.

No Noun Phrase Left Behind: Detecting and Typing Unlinkable Entities

In this paper they show that once the Wikipedia entities mentioned in a corpus of textual assertions are linked, this can further enable the detection and fine-grained typing of the unlinkable entities. There proposed method for detecting unlinkable entities achieves 24% greater accuracy than a Named Entity Recognition baseline, and our method for fine-grained typing is able to propagate over 1,000 types from linked Wikipedia entities to unlinkable entities. Detection and typing of unlinkable entities can increase yield for NLP applications such as typed question answering.
They introduce the unlinkable noun phrase problem: Given a noun phrase that does not link into Wikipedia, return whether it is an entity, as well its fine-grained semantic types. Deciding if a non-Wikipedia noun phrase is an entity is challenging because many of them are not entities.
The first part of this paper proposes a novel method for detecting entities by observing that entities often have different usage-over-time characteristics than non-entities. The second part of this paper shows how instance-to-instance class propagation can be adapted and scaled to semantically type general noun-phrase entities using types from linked entities, by leveraging over one million different possible textual relations.

Robust Disambiguation of Named Entities in Text

This paper presents a robust method for collective disambiguation, by harnessing context from knowledge bases and using a new form of coherence graph. It unifies prior approaches into a comprehensive framework that combines three measures: the prior probability of an entity being mentioned, the similarity between the contexts of a mention and a candidate entity, as well as the coherence among candidate entities for all mentions together. The method builds a weighted graph of mentions and candidate entities, and computes a dense sub graph that approximates the best joint mention-entity mapping.
Mention –Entity similarity measures used for Key phrase-based Similarity, Syntax-based Similarity.
The AIDA system provides an integrated NED method using popularity, similarity, and graph-based coherence, and includes robustness tests for self adaptive behavior. AIDA performed significantly better than state-of-the-art baselines. There future work will consider additional semantic properties between entities for further enhancing the coherence algorithm.

Entity Linking at Web Scale

This paper investigates entity linking over millions of high-precision extractions from a corpus of 500 million Web documents, toward the goal of creating a useful knowledge base of general facts. This paper describes new opportunities such as corpus-level features and challenges they found when entity linking at Web scale.

Information Extraction techniques such as Open IE operate at unprecedented scale. The REVERB extractor was run on 500 million Web pages, and extracted 6 billion extractions such as ("Orange Juice", "is rich in", "Vitamin C"), over millions of textual relations.

III. Methodology

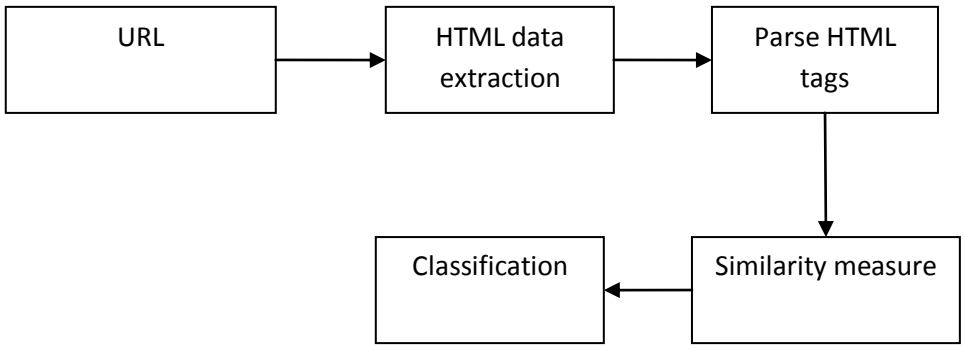


Figure 1: Block diagram

In the above figure 1 the URL's are taken as input and the data in the URL are extracted into plain text. After extraction the HTML tags are parsed. After converting each data into plain text than similarity measure is performed by using TF IDF and KNN clustering and then the classification of data is done by KMEANS. ***Calculate Term frequency***

Term Frequency can be calculate by simple formula that how many times the words is occurring in the document and calculating the frequency of the words and making another output file and store the term frequency.

Generally the term frequency is used to identify the unique keywords which are having less 'frequency' that keywords are highly used. So we can filter out the Keywords from the file of tokens with their relative position.

TF-IDF

The tf–idf weight (term frequency–inverse document frequency) is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. So we can figure out the importance of the keywords in the documents and then we can find out which keyword occurrence is there a in how many of the document with the help of IDF values have calculate from the formula. The main objective to IDF values, it can increases importance proportionally to number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf–idf weighting scheme are often used by search engines as central tool in scoring and ranking a relevant document's during query optimization phase when searching keywords in the search engine. ***Mathematical details***

The term count in the given document is simply the number of times a given term appears in that document. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term count regardless of the actual importance of that term in the document) to give a measure of the importance of the term t_i within the particular document d_j. Thus we have the term frequency, defined as follows.

$$\mathrm{tf_{i,j}} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Where $n_{i,j}$ is the number of occurrences of the considered term (t_i) in document d_j, and the denominator is the sum of number of occurrences of all terms in document d_j.

The inverse document frequency is a measure of the general importance of the term (obtained by dividing the number of all documents by the number of documents containing the term, and then taking the logarithm of that quotient).

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

with

- $|D|$: total number of documents in the corpus
- $|\{d : t_i \in d\}|$: Number of documents where the term t_i appears (that is $n_{i,j} \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to use $1 + |\{d : t_i \in d\}|$

Then

$$(\mathrm{tf\text{-}idf})_{i,j} = \mathrm{tf_{i,j}} \times \mathrm{idf_i}$$

A high weight in tf-idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. The tf-idf value for a term will always be greater than or equal to zero.

K-Means Clustering

When looking at data for the purpose of classification, there are several ways to approach classifying the examples in a given set. For example, we have parametric approaches, semi-parametric approaches, and nonparametric approaches.

As Ethem Alpaydin explains in Introduction to Machine Learning, "in the parametric approach, we assume that the sample comes from a known distribution." However, often that is simply not the case. In addition to the parametric methods, we also have nonparametric methods - which take the position that nothing can be assumed about the input density - or as Alpaydin puts it - "the data speaks for itself".

Somewhere in between these two opposite approaches lies the class of semi-parametric methods - those where we assume "a mixture of distributions," or "a parametric model for each group in the sample". It is under this umbrella that k-means clustering falls.

Simply put, k-Means Clustering is an algorithm among several that attempt to find groups in the data. In pseudo code, it is shown by Alpaydin (139) to follow this procedure:

Initialize $\mathbf{m}_i$, i = 1,…,k, for example, to k random $\mathbf{x}^t$

Repeat

For all $\mathbf{x}^t$ in X

$b_i^t \leftarrow 1$ if $\| \mathbf{x}^t - \mathbf{m}_i \| = \min_j \| \mathbf{x}^t - \mathbf{m}_j \|$

$b_i^t \leftarrow 0$ otherwise

For all $\mathbf{m}_i$, i = 1,…,k

$\mathbf{m}_i \leftarrow$ sum over t $(b_i^t \mathbf{x}^t)$ / sum over t (b_i^t)

Until $\mathbf{m}_i$ converge

The vector **m** contains a reference to the sample mean of each cluster. **x** refers to each of our examples, and **b** contains our "estimated [class] labels".

Explained perhaps more simply in words, the algorithm roughly follows this approach:

1) Choose some manner in which to initialize the m_i to be the mean of each group.
2) For each example in your set, assign it to the closest group (represented by m_i).
3) For each m_i, recalculate it based on the examples that are currently assigned to it.
4) Repeat steps 2-3 until m_i converge.

IV PROPOSED SYSTEM

K-Nearest Neighbor Classification

In the original k-nearest neighbor (KNN) classification method, no classifier model is built in advance. KNN refers back to the raw training data in the classification of each new sample. Therefore, one can say that the entire training set is the classifier. The basic idea is that the similar tuples most likely belongs to the same class (a continuity assumption). Based on some pre-selected distance metric (some commonly used distance metrics are discussed in introduction), it finds the k most similar or nearest training samples of the sample to be classified and assign the plurality class of those k samples to the new sample. The value for k is pre-selected. Using relatively larger k may include some pixels not so similar pixels and on the other hand, using very smaller k may exclude some potential candidate pixels. In both cases the classification accuracy will decrease. The optimal value of k depends on the size and nature of the data. The typical value for k is 3, 5 or 7. The steps of the classification process are:

1) Determine a suitable distance metric.
2) Find the k nearest neighbors using the selected distance metric.
3) Find the plurality class of the k-nearest neighbors (voting on the class labels of the NNs).
4) Assign that class to the sample to be classified.

We provided two different algorithms using P-trees, based two different distance metrics max (Minkowski distance with $q = \infty$) and our newly defined HOBS. Instead of examining individual pixels to find the nearest neighbors, we start our initial neighborhood (neighborhood is a set of neighbors of the target pixel within a specified distance based on some distance metric, not the spatial neighbors, neighbors with respect to values) with the target sample and then successively expand the neighborhood area until there are k pixels in the neighborhood set. The expansion is done in such a way that the neighborhood always contains the closest or most similar pixels of the target sample. The different expansion mechanisms implement different distance functions.

Of course, there may be more boundary neighbors equidistant from the sample than are necessary to complete the k nearest neighbor set, in which case, one can either use the larger set or arbitrarily ignore some of them. To find the exact k nearest neighbors one has to arbitrarily ignore some of them.

Instead we propose a new approach of building nearest neighbor (NN) set, where we take the closure of the k-NN set that is, we include all of the boundary neighbors and we call it the closed-KNN set. Obviously closed-KNN is a superset of KNN set. In the above example, with k = 3, KNN includes the two points inside the circle and any one point on the boundary. The closed-KNN includes the two points inside the circle and all of the four boundary points. The inductive definition of the closed-KNN set is given below.

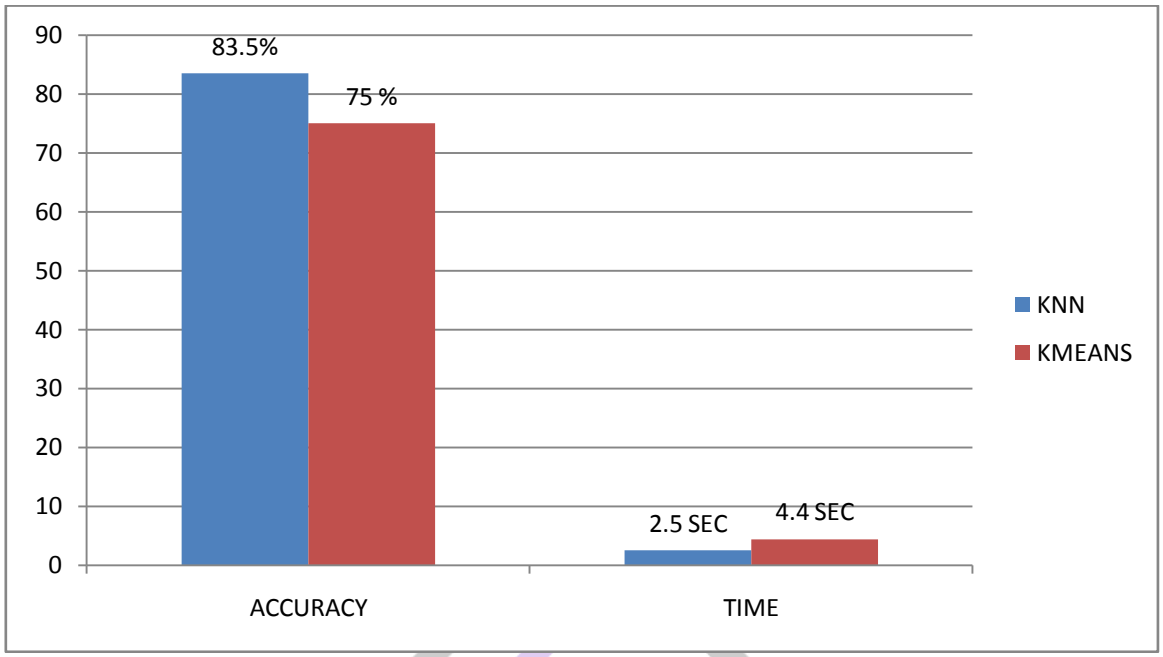
Definition 1:

a) if $x \in$ KNN, then $x \in$ closed-KNN

b) if $x \in$ closed-KNN and $d(T,y) \leq d(T,x)$, then $y \in$ closed-KNN Where, d(T,x) is the distance of x from target T.

c) closed-KNN does not contain any pixel, which cannot be produced by step a and b.

V. RESULTS



VI. CONCLUSION

We identified named entity classification as a particularly challenging task on Twitter. Due to their terse nature, tweets often lack enough contexts to identify the types of the entities they contain. In addition, a plethora of distinctive named entity types are present, necessitating large amounts of training data.

REFERENCES

[1] A. Sil, E. Cronin, P. Nie, Y. Yang, A.-M. Popescu, and A. Yates. Linking Named Entities to Any Database. In *EMNLP-CoNLL*, 2012.
[2] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP CoNLL*, pages 708–˝U716, 2007.
[3] Danuta Ploch.Exploring Entity Relations for Named Entity Disambiguation. Proceedings of the ACL-HLT 2011 Student Session, pages 18–23, Portland, OR, USA 19-24 June 2011.
[4] T. Lin, Mausam, and O. Etzioni. No Noun Phrase Left Behind: Detecting and Typing Unlinkable Entities. In *EMNLP*, 2012.
[5] J. Hoffart, M. A. Yosef, I. Bordino, H. Furstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum1. Robust Disambiguation of Named Entities in Text. In *EMNLP*, pages 782–792, 2011.
[6] T. Lin, Mausam, and O. Etzioni. Entity Linking at Web Scale. In *AKBC-WEKEX*, 2012.
[7] D. E. Brown, "The Regional Crime Analysis Program (ReCAP): a framework for mining data to catch criminals," SMC'98 Conf. Proceedings. 1998 IEEE Int. Conf. Syst. Man, Cybern. (Cat. No.98CH36218), vol. 3, pp. 2848–2853, 1998.
[8] L. Ding, D. Steil, M. Hudnall, B. Dixon, R. Smith, D. Brown, and A. Parrish, "PerpSearch: An integrated crime detection system," 2009 IEEE Int. Conf. Intell. Secur. Informatics, pp. 161–163, 2009.
[9] M. Alruily, A. Ayesh, and A. Al-Marghilani, "Using Self Organizing Map to cluster Arabic crime documents," Proc. Int. Multiconference Comput. Sci. Inf. Technol., pp. 357–363, Oct. 2010.
[10] B. Thuraisingham, "IEEE ISI 2008 Invited Talk (I) Data Mining for Security Applications : Mining Concept-Drifting Data Streams to Detect Peer to Peer Botnet Traffic," no. I, 2008.
[11] B. Chandra, M. Gupta, and M. P. Gupta, "A multivariate time series clustering approach for crime trends prediction," 2008 IEEE Int. Conf. Syst. Man Cybern., pp. 892–896, Oct. 2008.
[12] Y. L. Boo and D. Alahakoon, "Mining Multi-modal Crime Patterns at Different Levels of Granularity Using Hierarchical Clustering," 2008 Int. Conf. Comput. Intell. Model. Control Autom., pp. 1268–1273, 2008.
[13] P. Phillips, E. Peterphillipsjcueduau, and I. Lee, "Mining Top- k and Bottom- k Correlative Crime Patterns through Graph Representations," pp. 25–30, 2009.
[14] V. H. Bhat, P. G. Rao, A. R.V., P. D. Shenoy, V. K.R., and L. M. Patnaik, "A Novel Data Generation Approach for Digital Forensic Application in Data Mining," 2010 Second Int. Conf. Mach. Learn. Comput., pp. 86–90, 2010.
[15] W.-H. Chang and J.-S. Chang, "A Multiple-Phased Modeling Method to Identify Potential Fraudsters in Online Auctions," 2010 Second Int. Conf. Comput. Res. Dev., pp. 186–190, 2010.
[16] I. Conference and E. Technology, "2010 2nd International Conference on Education Technology and Computer (ICETC)," pp. 56–60, 2010.
[17] L. Ma, Y. Chen, and H. Huang, "AK-Modes: A weighted clustering algorithm for finding similar case subsets," 2010 IEEE Int. Conf. Intell. Syst. Knowl. Eng., pp. 218–223, Nov. 2010.
[18] Y. M. Lai, X. Zheng, K. P. Chow, L. C. K. Hui, and S. M. Yiu, "Automatic Online Monitoring and Data-Mining Internet Forums," 2011 Seventh Int. Conf. Intell. Inf. Hiding Multimed. Signal Process., pp. 384–387, Oct. 2011.