Engineering Applications of Artificial Intelligence 84 (2019) 11–17

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

SANE 2.0: System for fine grained named entity typing on textual data ☆

Anurag Lal, C. Ravindranath Chowdary *

Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, 221005, India

ARTICLE INFO

Keywords:
Named entity typing
Fined-grained
Wikipedia

ABSTRACT

Assignment of fine-grained types to named entities is gaining popularity as one of the major Information Extraction tasks due to its applications in several areas of Natural Language Processing. Existing systems use huge knowledge bases to improve the accuracy of the fine-grained types. We designed and developed SANE 2.0, which is an extended version of our earlier work SANE (Lal et al., 2017). It uses Wikipedia categories to fine grain the type of the named entities recognized in the textual data. The entities for which types could not be found using Wikipedia categories are typed using an intelligent information extraction method that uses search results of *yahoo* search engine. SANE uses an efficient algorithm to assign the fine-grained type to the entities extracted from the data. Wikipedia categorizes related topics under common headings. From these categories, we constructed a database that contains Wikipedia articles and their corresponding categories. SANE uses this database to predict the category types of named entities. We use Stanford NER to identify named entities with their coarse-grained types. For locations, we use Geonames data separately. We calculate the similarity between an entity and its categories using word2vec. Each entity is assigned to the category that has the highest similarity score with it. Finally, we map the category to the most appropriate WordNet (Miller et al., 1995) type. The main contribution of this work is building a named entity typing system without the use of knowledge bases. Through our experiments, 1) we establish the usefulness of Wikipedia categories to Named Entity Typing, 2) we present an intelligent method of using *yahoo* search results for Named Entity Typing and 3) we show that SANE's performance is on par with the state-of-the-art.

1. Introduction

Understanding the meaning of a text is the central problem in Natural Language Processing. Many sentences consist of one or more proper nouns that are people, locations and organizations. One popular approach to infer the meaning of any sentence is to assign the types to the entities in that sentence. These types help to establish the context in which the named entities are used. Named Entity Recognition (NER) is the task of identifying and classifying named entities in a sentence into pre-defined categories such as names of persons (PER), organizations (ORG), locations (LOC) or miscellaneous (MISC). If we use a finer granularity for typing the named entities, we would get more specific information about the entities. Tagging the entities with the fine-grained types makes the text corpus more structured and helps in its analysis.

In the task of Named Entity Typing we associate semantic types of interest with a given entity name. For instance, given "Sachin plays cricket", our objective is to conclude that "Sachin" is a *cricketer* or *sportsperson* and a *person*. In the attempt for a finer granularity, complex knowledge bases have been used like YAGO (Suchanek et al., 2007), DBPedia (Auer et al., 2007) and Freebase (Bollacker et al., 2008). These KB's use complex algorithms to populate and organize entities into semantic categories. However, the rate of generation, modification and termination of entities is very high. This accounts for the delay in the incorporation of such emerging entities in the knowledge bases, which makes knowledge bases inherently incomplete (Wang et al., 2012; Hoffart et al., 2014).

We propose SANE 2.0, a system that exploits Wikipedia's categorization to fine-grain this classification process of the Stanford NER. Named Entity Recognition is at the heart of natural language processing. It is closely associated with information extraction. The category labels for named entities are important in tasks such as machine translation (Babych, 2005) and question answering (Yahya et al., 2013). Also, entity types have proved to be useful in specialized areas such as bioinformatics (Rocktaschel et al., 2012) and molecular biology (Wattarujeekrit et al., 2004).

Explicit category types often occur in sentences where the category label itself is present with the entity in the input (Del Corro et al., 2015). These explicit category labels are often extracted via patterns,

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.engappai.2019.05.007.

* Corresponding author.
E-mail addresses: anurag.lal.cse13@iitbhu.ac.in (A. Lal), rchowdary.cse@iitbhu.ac.in (C. Ravindranath Chowdary).

https://doi.org/10.1016/j.engappai.2019.05.007
Received 7 June 2017; Received in revised form 10 February 2019; Accepted 24 May 2019
Available online xxxx
0952-1976/© 2019 Elsevier Ltd. All rights reserved.

like Hearst Patterns (Hearst, 1992). We have used a pattern-based extractor to extract such explicit category types. Wikipedia has been used to create knowledge bases (Hoffart et al., 2013) in the past. It has been used in NER by the authors in Bunescu and Pasca (2006), Toral and Munoz (2006), Kazama and Torisawa (2007). In the recent past, there has been a rapid growth of English Wikipedia, which now contains more than 5,000,000 articles (May 2016). As Wikipedia claims to be an encyclopedia, so most of the articles are about named entities. The advantage with Wikipedia is that new articles are being added frequently, so it keeps up with the dynamic scenario of new named entities appearing every day.

Wikipedia categories are used to group similar pages based on their content. They are implemented by a MediaWiki[1] feature which adds any page with a text like [[$Category : ABC$]] in its wiki markup, to the category with name ABC. Categories are usually found at the bottom of an article page. Clicking a category name will navigate to a category page (a page in the category namespace) which lists the articles that have been added to that category.

There are several entities for which Wikipedia pages do not exist. Intelligent web extraction methods can be used to alleviate this limitation of Wikipedia. A web search engine is a system that is used for searching information on the World Wide Web. A Search Engine Results Page (SERP) is the response of a search engine to a query of the search user. SERP usually contains two types of content: organic (as retrieved by the search engine algorithm) and sponsored (advertisements). Each organic result usually contains a title, a hyperlink to the actual Web page, and a brief description highlighting keywords that matched within the Web page. SANE 2.0 uses this description part of *yahoo*[2] search results for the purpose of NET.

1.1. Motivation

The majority of the recent work in this area uses complex knowledge bases to identify the most appropriate fine-grained entity type. However, knowledge bases take time to update. Also, since knowledge bases are very voluminous, exploiting them is resource intensive especially concerning memory. Recent approaches depend on huge knowledge bases for the accuracy of their results but update time of knowledge bases is greater than the update time of the data for entities dynamic in nature (Wang et al., 2012; Hoffart et al., 2014). Moreover, since the size of knowledge bases is much higher as compared to the size of corresponding databases, simple search queries but frequent in nature take lesser time against the database as compared to the corresponding knowledge base. Knowledge bases need periodic human curating to increase and maintain the accuracy of the KB (Deshpande et al., 2013). This motivated us to develop SANE 2.0, which uses $Wikipedia$ categories and *yahoo* search results to assign a fine-grained type to the named entity.

1.2. Our contributions

We use Wikipedia, which is the largest distributed encyclopedic database updated and monitored frequently. As compared to existing NET systems like FINET (Del Corro et al., 2015), Hyena (Yosef et al., 2013), our system SANE 2.0 is lesser memory intensive and does not depend on knowledge bases for type extraction. We demonstrate in this paper that category labels extracted from categories of a Wikipedia article and Web search results are useful to improve the granularity of NER. For example, "Banaras Hindu University" has the article associated with categories such as "Universities and colleges in Uttar Pradesh", "Educational Institutions established in 1916" and "Indian academics". These categories seem to be extremely useful for NER. We use such category labels to fine-grain the results of Stanford NER (Finkel et al., 2005). In our experiments, we used Twitter dataset to demonstrate that we can improve the granularity of the results of Stanford NER using Wikipedia categorization. We chose Twitter dataset for evaluation because tweets usually contain named entities for which articles, and thus categories, can be found in Wikipedia. To sum up the contributions:

1. We demonstrate the usefulness of Wikipedia categories for named entity typing
2. Our system incorporates only pattern-based and lookup-based extractors, which makes it simple as compared to the state-of-art systems
3. We present an intelligent method of using Web search results for NET

The rest of the paper is organized as follows. We first explain the working of SANE 2.0 in Section 2. In Section 3, we discuss the experimental results obtained by SANE 2.0 and possible causes of error on Twitter dataset. In Section 4, we discuss some related work that has been done in the past. Finally, we conclude our paper in Section 5.

2. Detailed description of SANE

In this section, we describe the procedure that SANE 2.0 uses for NET.

2.1. Generating database

As contrary to existing approaches of using large knowledge bases, we generated a database using Wikipedia categories. Wikipedia offers snapshots of all available content in the form of XML dumps.[3] We used the English version of Wikipedia at the time January 2014. Then we parsed the XML dump using Dizzy Logic Wiki Parser,[4] that generated files in plain text and XML formats. Using the parsed XML file, we extracted the topics with their corresponding categories into our database. The database contains 6,742,064 articles and 27,347,879 categories corresponding to these articles. We use this database in the lookup-based extraction phase for NET.

2.2. Generating location database

For locations, we created a database from Geonames,[5] consisting of 202 countries (including location aliases), 3893 states, and 41,023 cities. In a previous approach, we used Wikipedia categories for $LOCATION$ also but the quality of results was not good, so we created a static database from a comprehensive resource. We chose a static database because the names of geographical locations change less frequently. In a previous version of SANE (Lal et al., 2017), we used the Wikipedia for $LOCATION$ and we found that the Wikipedia categories are often pointing to the different contexts as compared to the current one, hence being irrelevant to NER. For 642 entities, we obtained a precision of 72% , but after using database derived from Geonames, the precision of SANE 2.0 increased to 80.82%.

2.3. Pattern-based extraction

Entity types are explicitly mentioned in the sentence usually to introduce the entities when they first appear in the text. In the first phase, SANE 2.0 looks for a set of patterns in the input sentence that explicitly refer to named entities. This is based on previous work by Hearst (1992). For example, if a sentence contains "Sachin Tendulkar, the cricketer", then the entity "Sachin Tendulkar" has the type "cricketer" explicitly mentioned. In this phase, such explicit types are extracted. We associate this type with the entity and directly go to

[1] https://mediawiki.org/wiki/MediaWiki.
[2] https://yahoo.com.
[3] https://dumps.wikimedia.org/enwiki/.
[4] https://dizzylogic.com/wikiparser.
[5] http://geonames.org/.

Table 1
Patterns used in SANE 2.0 (Del Corro et al., 2015).

Pattern	Example
Hearst	[Sachin] and other {cricketers}
Apposition	[Sachin], the {cricketer}
Copular	[Sachin] is a {cricketer}
Noun modifier	{Cricketer} [Sachin]

the type selection phase. This type is considered to be final, as it is explicit (Del Corro et al., 2015). If explicit categories are not found for a given entity, only then the next phase is entered.

We have used only four patterns that have high precision and do not lead to erroneous extractions. These patterns were chosen empirically. These patterns are listed in Table 1. Of these, the first pattern is called Hearst Pattern (Hearst, 1992).

2.4. Database lookup-based extraction

If explicit pattern-based extraction fails, then SANE 2.0 falls back to the lookup-based extraction. In this phase, SANE 2.0 exploits Wikipedia categorization by performing a lookup in a database containing article names and corresponding categories. We generated this database from the English version of Wikipedia XML dump in January 2014.[6] For *locations*, we created a database from Geonames,[7] consisting of 202 countries (including location aliases), 3893 states, and 41,023 cities. After SANE 2.0 performs the database lookup, in which the article titles are matched with the named entity and corresponding categories are added to a category list, it selects the best five categories based on our selection model. Also, SANE 2.0 uses Word2Vec (Mikolov et al., 2013) in the selection model to select the categories that have the highest average similarities to the context in which the named entity occurs. SANE 2.0 uses the similarity feature of Word2Vec for computing these average similarities. In a subsequent type selection phase, the explicit type from the pattern-based extraction phase or the categories selected in the lookup-based extraction phase are mapped to appropriate WordNet (Miller, 1995) types. Once the WordNet types are found, SANE 2.0 tags the named entity with the most appropriate type which is also selected based on our selection criteria. We also take into account the context in which the entity occurs in our similarity model.

SANE 2.0 performs a database lookup in which the article titles are matched with the named entity, and corresponding categories are added to a category list. The database lookup is said to be a success if at least one category type is added to the category list after the lookup. If the database lookup fails, then the system falls back to the web lookup-based extraction phase. This approach has a drawback that the database has to be maintained over time.

2.5. Web lookup-based extraction

In this phase, SANE 2.0 fetches the web page corresponding to *yahoo* search for the named entity that is to be typed. Then it parses the web page using the Beautiful Soup python library.[8] It then selects the description part of the search results which usually contains the type of the named entity. It then constructs a separate list of common nouns from the description part while discarding the ones that belong to a list of stop words. This list of common nouns is directly sent to the type selection phase.

[6] https://dumps.wikimedia.org/enwiki/.
[7] http://geonames.org/.
[8] https://pypi.python.org/pypi/beautifulsoup4.

2.6. Category selection

This phase selects most appropriate categories for a given named entity from the category list of that entity. Empirically, we found that selecting categories up to five yielded better results. In our experiments, we used gensim's python implementation of word2vec (Mikolov et al., 2013) and a pre-trained word2vec model trained on Google News.[9] This model has an inbuilt utility for finding the similarity between two words. For example, the similarity score between "bowled" and "cricket" is 0.38712 while the similarity between "deuce" and "cricket" is 0.03326.

Our category selection phase uses (1) the input sentence and the categories in the category list (2) the word2vec model. Category selection phase has three main stages: (1) Construction of lists of nouns and contexts, (2) Calculation of average similarity using word2vec semantic model, (3) Calculation of overall score for each category of every entity.

2.6.1. Construction of lists

For each category c_i in the category list of an entity e_s (entity e occurring in sentence s), we construct a list of common nouns occurring in that category. We call this list $Cat\_Nouns(c_i)$. We also take into account the context of s by generating the list $Context\_Nouns(s)$ that contains all the nouns present in multi-word named entities and the common nouns present in the sentence s.

2.6.2. Calculation of average similarity

To calculate the overall score, we first compute the average similarity between the nouns in $Context\_Nouns(s)$ and the nouns present in $Cat\_Nouns(c_i)$ using the procedure $avgsim(Context\_Nouns(s), Cat\_Nouns(c_i))$. Similarly, we compute the average similarity between the entity and the nouns present in $Cat\_Nouns(c_i)$ using $avgsim(e_s, Cat\_Nouns(c_i))$. We define $avgsim(A, B)$ as follows:

$$avgsim(A, B) = \frac{\sum_{a \epsilon A, b \epsilon B} similarity(a, b)}{(|A| * |B|)} \quad (1)$$

where A and B are lists of words and $similarity(a, b)$ is the word2vec similarity between a and b.

Algorithm 1: AVGSIM

```
avgsim(arg1,arg2);
Input: A,B
Output: Average similarity, as a real number between [0,1]
 A is a list of words;
 B is a list of words;
sim = 0;
l1 = A.size;
l2 = B.size;
for n1 in A do
    for n2 in B do
        sim += similarity(n1,n2);
return sim/(l1*l2);
```

2.6.3. Calculation of overall score

An overall score is computed for each category c_i in the category list of entity e_s. We add the average similarities $avgsim(Context\_Nouns(s), Cat\_Nouns(c_i))$ and $avgsim([e_s], Cat\_Nouns(c_i))$ to obtain the overall score for the category c_i. The expression for computing the overall score of a category is given in Eq. (2):

$$OS(c_i, e_s) = avgsim(Context\_Nouns(s), Cat\_Nouns(c_i)) + avgsim(e_s, Cat\_Nouns(c_i)) \quad (2)$$

In the above equation, $OS(c_i, e_s)$ is the overall score of the category c_i.

[9] https://drive.google.com/file/d/hellocheckthecommandB7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing.

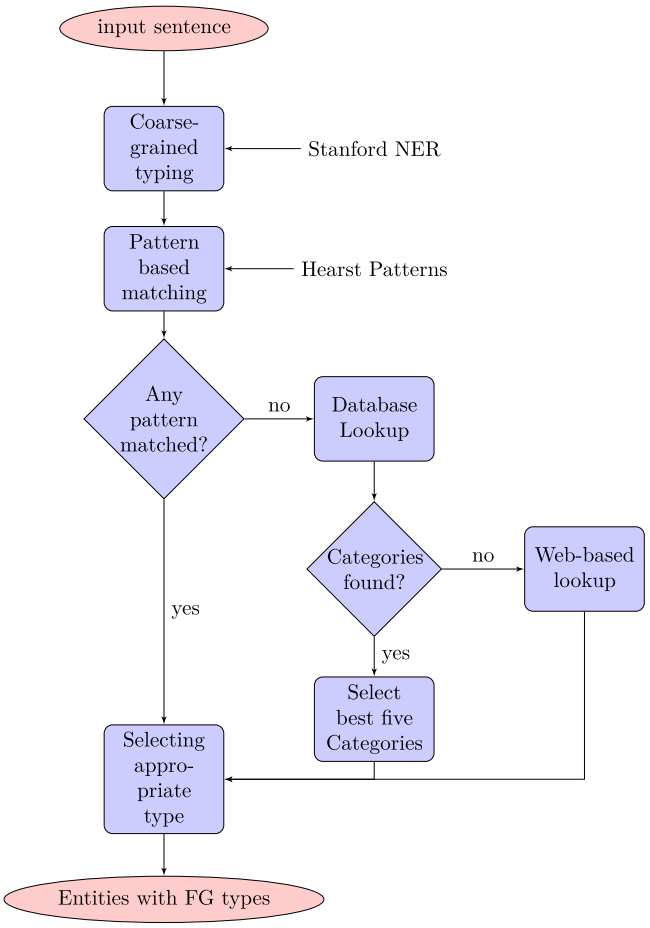

Fig. 1. Overview of SANE 2.0.

2.7. Type selection

In the type selection phase, we use the top five categories obtained from the category selection phase to collect the WordNet synsets and corresponding derivationally related forms (DER). From these synsets, we choose those synsets that are the hyponyms of the CG type of the corresponding entity. We use $< person - 1 >$, $< imaginary\_being - 1 >$ for $PERSON$ and $< organization - 1 >$ for ORGANIZATION. For each of these synsets, SANE 2.0 computes the average similarity between synset definition and the corresponding category as mentioned in Section 2.6.2. Finally, SANE 2.0 selects the synset having the maximum of the obtained similarities and associate this WordNet synset to the given named entity. The overview of SANE 2.0 is given in Fig. 1.

2.8. Example

For example, consider the sentence s "Banaras Hindu University is in Varanasi". This sentence is input to Stanford NER and output is the list of entities "Banaras Hindu University" and "Varanasi". Then for entity e_s "Banaras Hindu University", Stanford NER tags it coarsely as an $ORGANIZATION$. This entity e_s and sentence s are input to Hearst Pattern based matching. If there exists at least one pattern that matches the expression then we assume that the pattern-based matcher produced the fine grained entity type for the given e_s. Then we directly go to type selection phase. But in this example, e_s does not match any pattern. So, SANE 2.0 does a database lookup for e_s. If e_s occurs in the database, SANE 2.0 extracts the categories associated with e_s. In this case,suppose after the lookup-based extraction, the category list for the entity e_s "Banaras Hindu University", contains the categories c_1 "Universities and colleges in Uttar Pradesh" and c_2 "Educational Institutions established in 1916". The list $Cat\_Nouns(c_1)$ contains "universities" and "colleges" while $Cat\_Nouns(c_2)$ contains "institutions". $Context\_Nouns(s)$ contains "Banaras", "Hindu" and "University". Here, the value of $avgsim(Context\_Nouns(s), Cat\_Nouns(c_1))$ is 0.21905 while that of $avgsim(Context\_Nouns(s), Cat\_Nouns(c_2))$ is 0.14041 and the value of $avgsim([e_s], Cat\_Nouns(c_1))$ is 0.27354 while that of $avgsim([e_s], Cat\_Nouns(c_2))$ is 0.18513. Hence, the value of $OS(c_1, e_s)$ is 0.49259 while the value of $OS(c_2, e_s)$ is 0.32554. As there are less than five categories, we send both of them to the Type Selection phase. At this point, SANE 2.0 maps c_1 and c_2 to corresponding WordNet type. SANE 2.0 extracts common nouns "Universities", "colleges" and "institutions" from the categories for the given e_s. Then SANE 2.0 collects as candidate types the WordNet synsets that the categories refer. For category "Universities and colleges in Uttar Pradesh" synsets are <university-1>, <university-2> and <university-3> for noun "university" and <college-1>, <college-2> and <college-3> for noun "college". For category "Educational Institutions established in 1916", synsets are <institution-1>, <institution-2> and <institution-3>. Then SANE 2.0 computes average similarity of these synsets with corresponding categories. Similarity scores for these nouns are depicted in Table 2. Since the synset <university-1> has highest average similarity, SANE 2.0 associates it as a fine-grained type of e_s "Banaras Hindu University".

Table 2
Type selection.

Synsets	Average similarity score
<university-1>	0.16
<university-2>	0.07
<university-3>	0.08
<college-1>	0.15
<college-2>	0.10
<college-3>	0.08
<institution-1>	0.08
<institution-2>	0.08
<institution-3>	0.09

Table 3
Summary of results.

System	Total entities	Correct types	Precision
SANE 2.0 CG	3280	3177	96.86
FINET CG	3280	3177	96.86
SANE 2.0 FG	2540	2070	81.49
FINET FG	2154	1742	80.87

3. Experimental setup and results

We conducted experiments, using the Twitter dataset to compare SANE 2.0 with a state-of-the-art system FINET (Del Corro et al., 2015). FINET is a system for fine-grained typing of named entities in context. It makes use of multiple extractors for extracting both explicit and implicit types and then selects appropriate type in a subsequent type-selection phase. The type selection phase uses principles of Word Sense Disambiguation that are adjusted for fine-grained NER. It also leverages WordNet as its type system.

Data: We extracted 2000 tweets using the Twitter API. We specifically chose Twitter dataset as tweets usually contain entities that have Wikipedia articles. We selected only those tweets that contained at least one entity according to Stanford NER. We compared SANE 2.0 and FINET on this dataset.

System: SANE 2.0 uses the Stanford NER 3-class classifier to identify named entities with their coarse-grained category types in the tweets. We have classified the category types into coarse-grained (CG) and fine-grained (FG). SANE 2.0 processes the identified entities using the procedure outlined in Section 2. The CG category system consists

Table 4
SANE 2.0 extractor-wise performance.

Extractor	Entities	Precision
Pattern-based	55	90.91
Lookup-based	2485	81.28

```
Alan Turing is famous for the "Turing Test"
Alan Turing CG PERSON
Alan Turing FG philosopher
```

Fig. 2. FINET output for example 1.

```
Alan Turing is famous for the "Turing Test"
Alan Turing CG PERSON
Alan Turing FG cryptanalyst
```

Fig. 3. SANE 2.0 output for example 1.

```
Alex Ferguson rejected the chance to manage "Rangers"
Alex Ferguson CG PERSON
Alex Ferguson FG adult
```

Fig. 4. FINET output for example 2.

of three categories — PERSON, LOCATION, and ORGANIZATION while FG category system consists of hyponyms of $< person - 1 >$ and $< imaginary\_being - 1 >$ as $PERSON$, hyponyms of $< organization - 1 >$ as $ORGANIZATION$ and hyponyms of $< location - 1 >$ as $LOCATION$. FINET's FG type system consists of 200 WordNet types that are included in Pearl (Nakashole et al., 2013). FINET also generates super fine-grained (SFG) types that we are ignoring for comparison. Also, the precision of FINET for the SFG types is less than that for the FG types.

Labeling process: Two independent annotators label the results. We considered a category type to be correct if it was labeled correctly by both the annotators.

3.1. Performance analysis

The performance of FINET and SANE 2.0 is given in Table 3. For each system, the table shows the total number of entities for which category types were extracted, the number of correct category types and the precision for both CG and FG types. The Cohen's kappa measure[10] is 0.78 for FINET and 0.82 for SANE 2.0 indicating high inter-annotator agreement. For CG types, both SANE 2.0 and FINET used the Stanford NER, and hence have identical results. But SANE 2.0 assigned FG types to 2540 entities whereas FINET was able to assign FG types to only 2154 entities. This improvement can be attributed to the web-lookup based extraction phase that enables SANE 2.0 to categorize entities that were missed by the database-lookup extraction phase. FINET, on the other hand, does not use web-lookup for NET. Also, in spite of having a simpler design, SANE 2.0 has slightly more precision compared to FINET.

Table 4 shows the extractor-wise performance for FG types. We observe that the pattern-based extractor was able to assign FG types to only 55 entities but has a higher precision as compared to the lookup-based extractor (database and web-based lookup extractors together) which assigned FG types to 2485 entities. This vindicates our rationale for using the pattern-based extractor before the lookup-based extractor to assign FG types to entities.

[10] The Cohen's kappa measure is the overall score that includes both CG and FG types.

```
Alex Ferguson rejected the chance to manage "Rangers"
Alex Ferguson CG PERSON
Alex Ferguson FG football_player
```

Fig. 5. SANE 2.0 output for example 2.

3.2. Examples of NET

As can be seen from Fig. 2 FINET tags "Alan Turing" as *philosopher*, however from Fig. 3 SANE 2.0 tags "Alan Turing" as *cryptanalyst*. SANE 2.0 associates a finer type to person "Alan Turing". Similarly, we can see from Fig. 4, FINET tags "Alex Ferguson" as *adult* whereas SANE tags "Alex Ferguson" as *football_player* from Fig. 5. Here also, SANE assigns a finer type to person "Alex Ferguson".

3.3. Error analysis

The incompleteness of Wikipedia is a source of error. Some articles do not have any categories while some other articles have tens of categories making it difficult to select a category that fits into the context. Context can be taken into account in a better way by considering not only the words present in the sentence but also their lexical expansions. The category selection phase selects inappropriate categories for some entities as word2vec does not work very well with proper nouns. Due to the presence of incomplete names of entities, SANE 2.0 sometimes selects the wrong context. Misspellings of entity names and abbreviations are other sources of error.

4. Related work

In this section, we discuss the recent related work in the area of named entity recognition. Entity recognition and typing problem have been studied from perspectives of the degree of the context dependency and granularity (Ren et al., 2016). Traditional named entity recognition systems assume one to one mapping between the entity and the type and hence map entity recognition problem as multi-class classification. These systems take coarse-grained types e.g. $PERSON$, $ORGANIZATION$ and $LOCATION$ into consideration. The problem of NER has been addressed in many languages such as English, Spanish, Chinese, Japanese and Turkish (Kucuk and Yazıcı, 2012). In Konkol et al. (2015), authors propose novel features for NER based on latent semantics that are unsupervised. This enables language independent NER. In Jung (2012), authors propose contextual association properties to improve the performance of NER on microtexts in social networking services like Twitter.

Recent work has emphasized on a large set of fine-grained types. The size of fine-grained types varies from fixed to completely dynamic. A larger set of fine-grained types contradicts the assumption of one type per entity mention, converting the entity recognition problem into a multi-class multi-label classification problem. Some fine-grained systems use distant supervised techniques to train the examples while other systems use supervised embedding techniques to extract representative features. In Ren et al. (2016), authors propose to filter input data containing named entities to improve local context. However, there exist examples where words represent the correct context poorly, filtering which may change the context altogether.

Authors in Toral and Munoz (2006) and Kazama and Torisawa (2007) exploit Wikipedia as an external knowledge base for Named Entity Recognition. They focus on the first sentence of an article page that contains the definition of the entity described in the article and use it for NER. In Cucerzan (2007), authors use the information extracted from Wikipedia for recognition and semantic disambiguation of named entities. In Bunescu and Pasca (2006), authors use Wikipedia entity pages, redirection pages, categories and hyperlinks for disambiguation and built a context-article cosine similarity model and an SVM based

on a taxonomy kernel. SANE 2.0 solely uses Wikipedia categories for named entity typing.

FINET (Del Corro et al., 2015) addresses the NET problem by using a series of extractors from explicit to highly implicit relying on Yago2 (Suchanek et al., 2007) knowledge base. SANE 2.0 is very less memory intensive, as compared to FINET. Hyena (Yosef et al., 2013) is a supervised system of multi-label classification and faces problems of incorrect tagging occurring due to frequent co-occurrence of fine-grained types in KB. FIGER (Ling and Weld, 2012) considers entity-naming as a multi-label multi-class problem. They use anchor links in Wikipedia for information extraction. This approach has its drawback in the fact that there exist more than one anchor links belonging to a completely different context in the description of a given named entity. They provide a static set of overlapping entities curated from Freebase types. In Gillick et al. (2014), authors advocate the use of local context to determine the fine-grained type of the entity and uses a large set of trained mentions.

5. Conclusions

In this paper, we addressed the problem of Named Entity Typing (NET) efficiently. Our system SANE 2.0 uses Hearst Patterns, Wikipedia categories and open web for NET. We did not use knowledge bases which makes SANE 2.0 simple and scalable. We established the importance of Wikipedia categories to NET. Our experimental results show that SANE 2.0 is on par with a state-of-the-art system FINET. We also discussed the disadvantages of using knowledge bases and how our approach can address them.

5.1. Values and contributions

As compared to existing NET systems like FINET (Del Corro et al., 2015), Hyena (Yosef et al., 2013), our system SANE 2.0 does not depend on knowledge bases for type extraction. We demonstrate in this paper that category labels extracted from categories of a Wikipedia article are useful to improve the granularity of NER. For example, "Banaras Hindu University" has the article associated with categories such as "Universities and colleges in Uttar Pradesh", "Educational Institutions established in 1916" and "Indian academics". These categories seem to be extremely useful for NER. We use such category labels to fine-grain the results of Stanford NER (Finkel et al., 2005). In our experiments, we used Twitter dataset to demonstrate that we can improve the granularity of the results of Stanford NER using Wikipedia categorization. We utilized *yahoo* search results for the cases in which no categories are found in Wikipedia. To sum up the contributions:

1. We demonstrate the usefulness of Wikipedia categories for NET,
2. Our system incorporates only pattern-based and lookup-based extractors, which makes it simple as compared to the state-of-art systems,
3. We demonstrate an effective method of using *yahoo* search results for NET.

5.2. Limitations and future work

Performance of SANE 2.0 is heavily dependent on the categories of Wikipedia. Also, in web lookup, our system uses the descriptive part extracted by "yahoo". So, the performance is dependent on the quality of the descriptive part extracted by the retrieval system. As a future work, an ensemble of retrieval systems may be used to find the type of a named entity. Contents section of Wikipedia may also be explored along with categories for NET.

Acknowledgment

This work was partially funded under the DST-SERB, India grant YSS/2015/000906.

References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z., 2007. Dbpedia: A nucleus for a web of open data. In: Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference, ISWC'07/ASWC'07. Springer-Verlag, Berlin, Heidelberg, pp. 722–735.

Babych, B., 2005. IE methods for improving and evaluating MT quality. (Ph.D. Thesis). University of Leeds, Centre for Translation Studies.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J., 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08. ACM, New York, NY, USA, pp. 1247–1250.

Bunescu, R., Pasca, M., 2006. Using encyclopedic knowledge for named entity disambiguation. In: Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06), Trento, Italy, pp. 9–16.

Cucerzan, S., 2007. Large-scale named entity disambiguation based on wikipedia data. In: Proceedings of EMNLP-CoNLL, vol. 2007, pp. 708–716.

Del Corro, L., Abujabal, A., Gemulla, R., Weikum, G., 2015. Context-aware fine-grained named entity typing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Lisbon, Portugal, pp. 868–878.

Deshpande, O., Lamba, D.S., Tourn, M., Das, S., Subramaniam, S., Rajaraman, A., Harinarayan, V., Doan, A., 2013. Building, maintaining, and using knowledge bases: A report from the trenches. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, SIGMOD '13. ACM, New York, NY, USA, pp. 1209–1220.

Finkel, J.R., Grenager, T., Manning, C.D., 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In: T. A. for Computer Linguistics, (Ed.), Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363–370.

Gillick, D., Lazic, N., Ganchev, K., Kirchner, J., Huynh, D., 2014. Context-dependent fine-grained entity type tagging. arXiv preprint arXiv:1412.1820.

Hearst, M.A., 1992. Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th conference on Computational linguistics-, vol. 2, Association for Computational Linguistics, pp. 539–545.

Hoffart, J., Altun, Y., Weikum, G., 2014. Discovering emerging entities with ambiguous names. In: Proceedings of the 23rd International Conference on World Wide Web, WWW '14. ACM, New York, NY, USA, pp. 385–396.

Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G., 2013. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. Artificial Intelligence 194, 28–61.

Jung, J.J., 2012. Online named entity recognition method for microtexts in social networking services: A case study of twitter. Expert Syst. Appl. 39 (9), 8066–8070.

Kazama, J., Torisawa, K., 2007. Exploiting wikipedia as external knowledge for named entity recognition. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 698–707.

Konkol, M., Brychcín, T., Konopík, M., 2015. Latent semantics in named entity recognition. Expert Syst. Appl. 42 (7), 3470–3479.

Kucuk, D., Yazıcı, A., 2012. A hybrid named entity recognizer for turkish. Expert Syst. Appl. 39 (3), 2733–2742.

Lal, A., Tomer, A., Chowdary, C.R., 2017. Sane: System for fine grained named entity typing on textual data. In: Proceedings of the 26th International Conference on World Wide Web Companion, WWW '17 Companion. International World Wide Web Conferences Steering Committee, pp. 227–230.

Ling, X., Weld, D.S., 2012. Fine-grained entity recognition. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI'12. AAAI Press, pp. 94–100.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. CoRR abs/1301.3781.

Miller, G.A., 1995. Wordnet: A lexical database for english. Commun. ACM 38 (11), 39–41.

Nakashole, N., Tylenda, T., Weikum, G., 2013. Fine-grained semantic typing of emerging entities. In: ACL, vol. 1, The Association for Computer Linguistics, pp. 1488–1497.

Ren, X., El-Kishky, A., Wang, C., Han, J., 2016. Automatic entity recognition and typing in massive text corpora. WWW '16 Companion, Republic and Canton of Geneva, Switzerland, pp. 1025–1028.

Rocktaschel, T., Weidlich, M., Leser, U., 2012. Chemspot: A hybrid system for chemical named entity recognition. Bioinformatics 28 (12), 1633–1640.

Suchanek, F.M., Kasneci, G., Weikum, G., 2007. Yago: A core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, WWW '07. ACM, New York, NY, USA, pp. 697–706.

Toral, A., Munoz, R., 2006. A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia. In: EACL 2006.

Wang, S., Li, C., Chen, H., 2012. A conversation with professor Shan Wang et al. SIGKDD Explor. Newsl. 13 (2), 92–95.

Wattarujeekrit, T., Shah, P.K., Collier, N., 2004. Pasbio: Predicate-argument structures for event extraction in molecular biology. BMC Bioinformatics 5 (1), 155.

Yahya, M., Berberich, K., Elbassuoni, S., Weikum, G., 2013. Robust question answering over the web of linked data. In: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, CIKM '13. ACM, New York, NY, USA, pp. 1107–1116.

Yosef, M.A., Bauer, S., Hoffart, J., Spaniol, M., Weikum, G., 2013. Hyena-live: Fine-grained online entity type classification from natural-language text. In: ACL (Conference System Demonstrations) 2013. The Association for Computer Linguistics, pp. 133–138.