Named Entity Extraction Using Information Distance*

Sangameshwar Patil†, Sachin Pawar‡, Girish K. Palshikar
Tata Research Development and Design Centre, TCS
54B, Hadapsar Industrial Estate, Pune 411013, India
{sangameshwar.patil,sachin7.p,gk.palshikar}@tcs.com

Abstract

Named entities (NE) are important information carrying units within documents. *Named Entity extraction (NEX)* task consists of automatic construction of a list of phrases belonging to each NE of interest. NEX is important for domains which lack a corpus with tagged NEs. We present an enhanced version and improved results of our unsupervised (bootstrapping) NEX technique (Patil et al., 2013) and establish its domain independence using experimental results on corpora from two different domains: agriculture and mechanical engineering (IC engine[1] parts). We use a new variant of Multiword Expression Distance (MED) (Bu et al., 2010) to quantify proximity of a candidate phrase with a given NE type. MED itself is an approximation of the information distance (Bennett et al., 1998). Efficacy of our method is shown using experimental comparison with pointwise mutual information (PMI), BASILISK and KNOWITALL. Our method discovered 8 new plant diseases which are not found in Wikipedia. To the best of our knowledge, this is the first use of NEX techniques for agriculture and mechanical engineering (engine parts) domains.

1 Introduction

Agriculture is an activity of fundamental importance in all societies. For example, agriculture plays a vital role in the economy of India: generating second largest farm output in the world, providing 52% of rural employment ($\approx$ 250 million people) and contributing $\approx 15\%$ to the GDP [2]. Hence timely and widespread dissemination of agriculture-related information is important. Such information - extracted, collated and summarized from a variety of document sources such as news, reports, web-sites and scientific literature - can be used to improve various aspects of services provided to large population dependent on agriculture. Problem of information extraction for domains such as agriculture is particularly challenging due to non-availability of any tagged corpus.

Several domain-specific *named entities (NE)* occur in the documents (such as news) related to the agriculture domain- CROP: names of the crops including varieties; DISEASE: names of the crop diseases and disease causing agents such as pathogen (bacteria, viruses, fungi), insects etc.; CHEMICAL_TREATMENT: names of pesticides, insecticides, fungicides etc. used in the treatment of a crop disease. Consider an example of NEs tagged for agricultural information extraction:

We usually spray [soybeans]$_{CROP}$ with a [strobilurin fungicide]$_{CHEM\_TREATMENT}$ because of the potential for [soybean rust]$_{DISEASE}$ and other diseases.

As there are few, if any, tagged corpora of agriculture-related documents, we consider an unsupervised approach for extracting these NEs. *NE extraction (NEX)* problem consists of automatically constructing a gazette containing example instances for each NE of interest. A *NE recognition (NER)* algorithm basically matches the given gazette G (for a particular NE, say DISEASE) with the given document D to identify occurrences of the instances from G in D. While gazette-based NER is fast, the accuracy depends on the quality and completeness of the gazette.

In this paper, we present an enhanced version

*Preliminary version of this paper was presented as a poster at NLDB 2013.

†Doctoral research scholar at Dept. of CSE, IIT Madras

‡Doctoral research scholar at Dept. of CSE, IIT Bombay

[1]Internal Combustion engine

[2]http://en.wikipedia.org/wiki/Economy_of_India (access date 31-Jan-2013)

International Joint Conference on Natural Language Processing, pages 1264–1270,
Nagoya, Japan, 14-18 October 2013.

and improved results of our bootstrapping approach to NEX (Patil et al., 2013) and establish its domain independence. We demonstrate the use of this NEX technique for creating gazettes of NE in the agriculture and mechanical engineering domains. Apart from the new application domains for NE extraction, other specific contributions of this paper are as follows: We propose a new variant of the well-known information distance (Bennett et al., 1998; Li et al., 2004; Bu et al., 2010) measure, to decide whether a candidate phrase is a valid instance of the NE or not. We use additional tools and show their effectiveness on improving the gazette quality: (i) a candidate generation algorithm based on maximum entropy classifier; and (ii) a post processing algorithm in the spirit of the assessor module in (Etzioni et al., 2005), but we use statistical hypothesis-testing. Utility and effectiveness of our method is evident from its ability to discover new named entities. For instance, using a limited news corpus, we discovered 8 new crop diseases which are not mentioned in Wikipedia.

The rest of the paper is organized as follows. Section 2 contains a brief overview of related work. In Section 3, we summarize information distance and multiword expression distance, along with new extensions. In Section 4, we discuss our unsupervised algorithm for gazette creation. In Section 5, we present experimental results. Finally we conclude with an outline of future work.

2 Related Work

Thelen and Riloff (2002) propose a bootstrapping algorithm called BASILISK for NEX. Etzioni et al. (2005) present a system called KNOWITALL, which implements an unsupervised domain-independent, bootstrapping approach to generate large facts of a specified NE (such as CITY or FILM) from the Web. Many other unsupervised approaches have been proposed for both NEX and NER: (Collins and Singer, 1999; Kim et al., 2002; Meulder and Daelemans, 2003; Talukdar et al., 2006; Jimeno et al., 2008; Liao and Veeramachaneni, 2009; Palshikar, 2012). The basic structure of the bootstrapping approach to NEX is well-known: starting with a seed list of examples of a particular NE type, iteratively identify other phrases which are "similar" to them and add them to the seed list. The algorithm terminates either after a specified number of iterations, or on reaching a specified gazette size, or when no new entries get added or when the new entities show a "drift".

3 Information Distance for NE

Information distance (Bennett et al., 1998) is an abstract and universal domain-independent, statistically motivated distance measure based on the concept of *Kolmogorov complexity*. Given a Universal Turing Machine (UTM) U, *Kolmogorov complexity* $K_U(x|y)$ of a binary string x, given another binary string y, is the length of the shortest program for U that computes x when given y as input. Bennett et al. (1998) showed that the quantity (which they called the *information distance*) $D_{max}(x, y) = max\{K(x|y), K(y|x)\}$ measures the distance between objects (such as binary strings) x and y.

Bu et al. (2010) presented a variant of the information distance, which measures the distance between an n-gram and its semantics. They define the *context* $\phi(g)$ of an n-gram g as the set of all web-pages containing g while the *semantics* $\mu(g)$ of g is the set of all web-pages that contain all words in g but not necessarily as a contiguous n-gram. For example, for g = `Bill Gates`, $\phi(g)$ consist of all pages containing `Bill Gates` as the bigram while $\mu(g)$ consists of all web-pages that contain both `Bill` and `Gates` but not necessarily as a bigram. Clearly, $\phi(g) \subseteq \mu(g)$. The *Multiword Expression Distance (MED)* $MED(g)$ measures the distance of g from an "intended" (non-compositional) semantics:

$$MED(g) = D_{max}(\phi(g), \mu(g))$$

$$MED(g) \approx log|\mu(g)| - log|\phi(g)|$$

Bu et al. (2010) demonstrated the use of MED to perform NER. In this paper, we use a variant of MED to perform NEX. Unlike (Bu et al., 2010), we are constrained to use a corpus rather than the entire Web. Let $\mathbf{D}$ be a given untagged corpus of sentences. Let K be a given constant indicating the window size (e.g., $K = 3$). Let g be a given candidate phrase. The *context* of g and a given word w, denoted $\phi_K(g, w)$, is the set of all sentences in $\mathbf{D}$ which contain both g (as a n-gram) as well as w and further, w occurs within a window of size K around g in that sentence. The *semantics* of g and a given word w, denoted $\mu(g, w)$, is the set of all sentences in $\mathbf{D}$ which contain both g (as an n-gram) and w, though g and w need

not be within a window of size K in the sentence. Clearly, $\phi_K(g, w) \subseteq \mu(g, w)$. Then we define the *distance* between g (a given candidate phrase) and a given word w as follows:

$$MED0_{D,K}(g, w) = log|\mu(g, w)| - log|\phi_K(g, w)|$$

Let $W = \{w_1, w_2, \ldots, w_m\}$ be a given finite, non-empty set of m words. The definition of $MED0$ is extended to use a given set W (rather than a single word w) by taking the average of the $MED0$ distance between g and each word in W:

$$MED_{D,K}(g, W) =$$

$$\frac{MED0_{D,K}(g, w_1) + \ldots + MED0_{D,K}(g, w_m)}{m}$$

We assume that a subroutine $MED(\mathbf{D}, K, W, g)$ returns the MED distance $MED_{D,K}(g, W)$, as defined above.

4 NEX Using MED

In NEX task, we are given (i) an untagged corpus $\mathbf{D}$ of documents; and (ii) a seed list L containing known examples of a particular NE type T. The goal is to create a gazette containing other instances of the NE type T that occur in $\mathbf{D}$. We first look at some sub-problems and then give the complete algorithm for NEX in Section 4.4.

4.1 Pre-processing step

The first task is to identify all phrases in $\mathbf{D}$ that are likely to be instances of the NE type T. In the simplest case, all phrases in $\mathbf{D}$ that have the same syntactic structure as the instances in L could be considered as candidates. For example, for DISEASE, one could look for all NPs that may begin with at most one adjective (e.g., `gray mold`), followed by one or more nouns and whose head word is a singular common noun (as in `crown rot` or `tissue blight`). We assume such simple logic is implemented in a subroutine $GenCandidates$ (not shown in this paper). However, even with such syntactic restrictions, the number of candidate instances is usually very large. For instance, for agricultural domain corpus described in Section 5, around 350,000 candidates were output by $GenCandidates$. It also contains many phrases which are obviously not the instances of T (e.g., `moderate field tolerance`).

Algorithm $Prune$ (Fig. 1) *pre-processes* the list C of candidates from $GenCandidates$ using a self-trained, iterative maximum entropy (MaxEnt) classifier. Some of the major features used by MaxEnt classifier are: words in the phrase, words in context and their POS tags, next and previous verbs, binary features to capture presence of adjective, capitalization, proper nouns within phrase.

function Prune($\mathbf{D}$, C, n_0, L_1, L_2)
$G_1 := \emptyset$; $G_2 := \emptyset$;
for i :=0; $i <$ *MaxIter*; i++ **do**
  $H_1 := \emptyset$; $H_2 := \emptyset$;
  Build maximum entropy classifier M using instances in $L_1 \cup G_1$ and $L_2 \cup G_2$ as positive and negative examples for class T respectively;
  foreach phrase $g \in C$ && $g \notin G_1 \cup G_2$ **do**
    (c, pc) := predicted class of g using M along with predicted probability for class c
    if $c ==$ T **then** $H_1 := H_1 \cup \{(c, pc)\}$
    else $H_2 := H_2 \cup \{(c, pc)\}$ **endif**
  end foreach
  Retain only top n_0 elements in H_1 and in H_2 in descending order of pc values
  $G_1 = G_1 \cup H_1$ // add only the phrases from H_1 to G_1
  $G_2 = G_2 \cup H_2$ // add only the phrases from H_2 to G_2
end for
return (G_1)

Figure 1: Algorithm *Prune*

4.2 Backdrop of a Gazette

The key problem in unsupervised NEX is to decide, using $\mathbf{D}$, whether a candidate phrase g has the same NE type T as the examples in L. For example, given L = {`gray mold`, `crown rot`, `tissue blight`} as a seed list for T = DISEASE and g = `corn rust` as a candidate phrase, we need to decide whether g is a DISEASE or not. The idea is to use the $MED_{D,K}$ as defined above to accept only those g which have "low" distance ("high" similarity) between g and the backdrop of the gazette L, which is defined as a set W of words "characteristic" (or strongly indicative) of T.

Function $GetBackdrop(\mathbf{D}, L, K, m_0)$ (Fig. 2) computes, using $\mathbf{D}$, the set W for a given gazette L. Essentially, it computes how many times each word w occurs in the context window of given size K around every entry in L. Then it computes a *relevance score* for each word. The *relevance score* is computed as a product of following factors: the entropy H of the word; the number b of entries in L for which it is a context word; and ratio of the total number f_L of times the word occurs in the context of all entries in L to its frequency f in the entire corpus $\mathbf{D}$. Finally, it returns the top m_0 words in terms of the highest relevance score values as the backdrop for

function GetBackdrop(**D**, L, K, m_0)
$W = \emptyset$ // initially empty
$h = \emptyset$ // hash table key=word value=count
foreach word w in **D** **do**
  foreach entry $u_i \in L$ **do**
    compute the frequency $f(w,u_i)$ of how many times w occurs in the context window of size K for u_i
  end foreach
  // $f(w)$ = total no. of occurrences of w in **D**
  // $f_L(w) = f(w,u_1)+f(w,u_2)+\ldots+f(w,u_{|L|})$ = total no. of occurrences of w in the context of all entries in L
  compute the entropy of w as
  $$H(w) = -\sum_{i=1}^{|L|} \frac{f(w,u_i)}{f_L(w)} \log \left(\frac{f(w,u_i)}{f_L(w)} \right)$$
  $b(w)$:= no. of entries $u_i \in L$ for which $f(w,u_i) > 0$
  Define $score(w) := f_L(w)/f(w) \times b(w) \times H(w)$
end foreach
W := select top m_0 words in terms of their scores
return(W)

Figure 2: Algorithm $GetBackDrop$

L. For example, suppose L consists of 6 DISEASE instances. For the word `causes`, b(`causes`) = 6, H(`causes`) = 1.51 and f(`causes`) = 75, leading to $score$(`causes`) = 16.9895. For the word `technology`, b(`technology`) = 2, H(`technology`) = 0.0646 and f(`technology`) = 84, leading to $score$(`technology`) = 0.2485. Clearly, `causes` is much more relevant as a cue word for DISEASE than `technology`. The score gives more importance to words that appear more frequently as well as in the context of more entries in the given gazette. Higher entropy value indicates that the word is used more uniformly in the context of many entries in L. Words with a more skewed usage (lower entropy) may be good indicator words for specific entries rather than for all entries in L. Such words are not preferred.

4.3 Assessing the Gazette

We propose a *post-processing* step to assess and improve the quality of the candidate gazette created, by identifying (and removing) those entries in the candidate gazette which are very unlikely to be true instances of NE type T.

Suppose the candidate gazette includes the two phrases g_1 = `late blight` and g_2 = `wet weather`. Suppose we had also been given a small set Q of *cue words* for the NE type T; e.g., for DISEASE, Q could be {`disease, cause, symptom`}. For a given phrase g, we compute two counts: $c(g)$ = count of sentences in **D** which contain the n-gram g and $cq(g)$ = count of sentences in **D** which contain (in any order) both the n-gram g and at least one word in Q. Clearly, $cq(g) \leq c(g)$. Let $\hat{f}(g) = \frac{cq(g)}{c(g)}$. Clearly, $0 \leq \hat{f}(g) \leq 1$. For example, $\hat{f}(g_1) = 38/90 = 0.422$ and $\hat{f}(g_2) = 104/642 = 0.162$.

Let $0 < f_0 < 1$ be a fixed value; e.g., $f_0 = 0.2$ (we shall shortly discuss how to obtain f_0). Essentially, $\hat{f}(g)$ indicates how well the phrase g "co-occurs" with words in Q. "Low" values of $\hat{f}(g)$ (e.g., those below f_0) indicate that the number of occurrences of g drops drastically when you restrict to only those sentences that contain at least one word in Q. Such phrases are unlikely to be true instances of T. We perform a statistical hypothesis test (called *proportion test*) that the fraction $\hat{f}(g)$ is greater than the given constant f_0. The null hypothesis is $H_0 : f(g) \geq f_0$, where $f(g)$ is the "true" proportion for g, since the observed proportion varies depending on **D**. The test statistic is

$$\frac{\hat{f}(g) - f_0}{\sqrt{\frac{f_0(1-f_0)}{c(g)}}}$$

which follows the Standard Normal distribution. Hence the probability (p-value) of observing a particular value of the test statistic can be computed using standard tables. The null hypothesis is rejected if p is less than the given significance level α (we use $\alpha = 0.05$).

For g_1, g_2, the test statistic values are 5.270 and -2.407; and the p-values are 0.999 and 0.008. Thus g_2 is correctly rejected as an unlikely instance for the NE type DISEASE. Note that this step requires the user to provide the set Q of cue words for NE type T; we found that generally only a few words (≤ 10) are enough. We set f_0 to just below the minimum among the values for the current gazette L. We assume that this logic is implemented as a function $Assessor(\mathbf{D}, Q, L)$, where L is the gazette containing phrases to be assessed.

4.4 Unsupervised Gazette Creation

Algorithm $CreateGazetteMED$ (Fig. 3) coordinates various modules described so far in this section. It starts with an initial seed list L of instances of a particular NE type T. It first calls the algorithm $GenCandidates$ to create a list C of candidate phrases for T using **D** and prunes C using the algorithm $Prune$. Then in each iteration, it calls the algorithm $GetBackdrop$ to cre-

```
algorithm CreateGazetteMED
input  D // set of all sentences form the corpus
input  L = {g_1, ..., g_n} // seed list of NE instances
input  L_2 // seed list of entity non-instances
input  Q // set of cue words for NE type T
input  K // context window size; default = 3
input  n_0 // no. of candidate instances; default 50000
input  h_0 // threshold for MED; default = 0.2
input  m_0 // no. of backdrop words; default = 150
input  maxIter // maximum no. of iterations; default = 15
output L // gazette with new entries added
C := GenCandidates(D)
C := Prune(D, C, n_0, L , L_2)
for i = 1; i < maxIter; i++ do
   A := ∅ // initially empty
   W := GetBackdrop(D, L, K, m_0)
   foreach candidate phrase g ∈ C && g ∉ L do
      if MED(D,K,W,g) ≤ h_0 then
         A := A ∪ {g}
      endif
   end foreach
   L := L ∪ A // add entries in A to L
end while
L := Assessor(D,Q,L) // remove unlikely entries
```

Figure 3: Algorithm $CreateGazetteMED$

ate the set W of backdrop words for T using L. Then it uses the modified MED to measure the similarity of each candidate phrase $g \in C$ with W and adds g to a temporary set A only if it has a "high" similarity with W (above a threshold). A configurable number (default 10) of top candidates from set A are added to L in each iteration. At the end of $maxIter$, final set of candidates in L is then pruned using the $Assessor$ algorithm. We have enhanced the earlier version (Patil et al., 2013) by using weighted backdrop to compute $MED_{D,K}(g, W)$. Note that the algorithm contains four independent ways of assessing whether a sequence of words is an instance of T or not, as implemented in $GenCandidates$, $Prune$, $MED_{D,K}$ and $Assessor$.

5 Experimental Evaluation

Experimental Setup: In addition to the agricultural news corpus described in (Patil et al., 2013), we also evaluated the proposed technique on a corpus of 7500 engine repair records from mechanical engineering domain. Goal is to create a gazette of engine part names from the engine repair records. The agriculture corpus consists of 30533 documents in English containing 999168 sentences and approximately 19 million words. It was collected using crawler4j (Ganjisaffar, 2013) by crawling the agriculture news websites (websites, 2012)[3]. A sample of seeds used to bootstrap the NEX for each category are as following - CROP: {`wheat, cotton, corn, soybean, banana`} ; DISEASE: {`wilt, leaf spot, rust, weevil`} ; CHEM_TREATMENT: {`di-syston, evito, tilt, headline`} ; ENGINE_PARTS:{`piston, gaskets, bearings, crankshaft, cylinder`}.

	Crop	Disease	Chem. Treatment	Engine Part
$MED_{D,K}$ with assessor	352 (0.662)	237 (0.928)	332 (0.886)	88 (0.818)
PMI with assessor	419 (0.625)	315 (0.911)	341 (0.883)	92 (0.793)
$MED_{D,K}$ no assessor	372 (0.637)	267 (0.831)	502 (0.671)	91 (0.802)
PMI no assessor	441 (0.603)	361 (0.801)	512 (0.670)	95 (0.779)
BASILISK	352 (0.278)	237 (0.924)	332 (0.386)	100 (0.67)

Figure 4: Number of entries (& precision) in the final gazette for each NE type. (To use the same baseline for comparing precision of the proposed algorithm and BASILISK, we use the gazette size of BASILISK comparable to that of $MED_{D,K}$ with Assessor.)

Results: Fig. 4 summarizes the gazette sizes along with precision for NE types from both agriculture and mechanical engineering (IC engine parts) corpora. Detection rate for agriculture NE type are shown in Fig. 5. To calculate the precision, all the gazettes created by all the algorithms were manually and independently verified by at least three different human annotators. From these results, we conclude that the proposed technique performs well in *domain independent* manner. Assessor module improves precision for all NE types for both measures $MED_{D,K}$ and PMI. Post-processing using assessor has positive impact on gazette quality. In this experiment, we used top 1000 candidates produced by algorithm $Prune$, instead of top 5000 used in the earlier version (Patil et al., 2013). We observe significant improvement in precision for all NE types. It is clear that the gazette quality is dependent on the output of algorithm $Prune$. We are investigating their inter-relationship as part of this on-going work.

A sample entries from gazette created for each NE type T are as follows: CROP: {`rr alfalfa, sugarcane, biofuel crops, winter canola`};

[3]Permission awaited from the content-owners for public release of the corpus for research purpose.

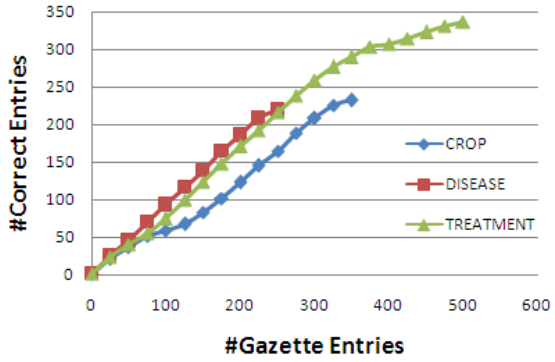

Figure 5: Detection rate of $CreateGazetteMED$ with Assessor

DISEASE: { `asian soybean rust, alternaria, downy mildew, citrus greening, fusarium wilt`}; CHEM_TREATMENT: { `ultra blazer, strobilurin, telone II, gaucho grande, spartan`}; ENGINE_PARTS:{ `piston pin, conn rod, cam gear, cyl head, piston skirt`}.

To highlight effectiveness of the gazettes created, we compared our DISEASE gazette with wikipedia. We listed all the page titles on wikipedia falling under categories *Plant pathogens and diseases* (1935) and *Agricultural pest insects* (203). It was quite encouraging to find that, our gazette, though created on a limited size corpus, contained diseases/pathogens not present in wikipedia.[4] Some of these are - `limb rot, grape colaspis, black shank, glume blotch, seed corn maggot, mexican rice borer, green bean syndrome, hard lock`.

Comparison with BASILISK: Our implementation of BASILISK (Pawar et al., 2012) has comparable precision with the proposed $CreateGazetteMED$ algorithm for DISEASE category. However, $CreateGazetteMED$ clearly outperforms BASILISK for all other categories. Major reason behind worse performance of BASILISK is that it scores each occurence of the phrase in the corpus independently and the decision to add that phrase to the gazette depends on the highest among all of these scores. $CreateGazetteMED$ computes a single score for each phrase by combining evidences from all of its occurrences in the corpus.

Comparison with KNOWITALL: KNOWITALL (Etzioni et al., 2005) is a leading, web-scale unsupervised information extraction engine. We implemented basic version of KNOWITALL algorithm and executed it on above mentioned corpus of agricultural news items. For agriculture domain, KNOWITALL extracted gazettes of following sizes for the three NE types - CROP : 36 entries (20 correct); DISEASE : 55 entries (49 correct); CHEM_TREATMENT : 13 entries (12 correct).

We believe that reason for KNOWITALL's limited gazette size lies in inherent difference between a web-scale search vis-a-vis searching a given corpus. This results in skewed search query statistics and affects the size of gazettes created.

Comparison with PMI: To gauge the effictiveness of $MED_{D,K}$ as a proximity measure, we compare it with PMI (Bouma, 2009). We follow exactly the same steps as described in our $CreateGazetteMED$ algorithm with the only difference being use of PMI, instead of $MED_{D,K}$. For the results with top 1000 candidates from Prune (Fig. 4), $MED_{D,K}$ compares favorably with PMI as a proximity measure for all the NE types. Comparing with results with top 5000 candidates from Prune (in (Patil et al., 2013)), we observe that $MED_{D,K}$ is a more robust proximity measure than PMI. Sensitivity of these measures to output of pre-processing step is part of future work.

6 Conclusions and Further Work

We presented improved results of our unsupervised NEX technique, $CreateGazetteMED$. A new variant of MED is used to quantify the proximity (similarity) of a candidate phrase with a given NE type. We estabhlished its domain independence using corpora from agriculture and mechanical engineering domains. Effectiveness of $CreateGazetteMED$ was validated using experimetal comparison with PMI, BASILISK and KNOWITALL. Our method incorporated a preprocessing step (based on MaxEnt classifier). We also proved efficacy of statistical hypothesis testing as a post-processing step to improve gazette quality. As part of further work, developing a robust stopping criterion for automatically stopping the gazette creation process needs attention. Unsupervised relation extraction (such as relations between CROP, DISEASE, CHEMICAL_TREATMENT and many other NEs) is a natural extension. Establishing language independence of the proposed technique, exploring effect of number and quality of initial seeds are also promising avenues.

[4] Verified on 30^{th} January, 2013

References

C.H. Bennett, P. Gacs, M. Li, P.M.B. Vitanyi, and W.H. Zurek. 1998. Information distance. *IEEE Transactions on Information Theory*, 44(4):1407–1423.

G. Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*.

F. Bu, X. Zhu, and M. Li. 2010. Measuring the non-compositionality of multiword expressions. In *Proceedings Of the 23rd Conf. on Computational Linguistics (COLING)*.

M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP)*.

O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, and D.S. Weld. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence*, 165:91–134.

Yasser Ganjisaffar. 2013. `http://code.google.com/p/crawler4j/`. [Online; accessed 16 Aug. 2013].

A. Jimeno, E. Jimenez-Ruiz, V. Lee, S. Gaudan, R. Berlanga, and D. Rebholz-Schuhmann. 2008. Assessment of disease named entity recognition on a corpus of annotated sentences. *BMC Bioinformatics*, 9(Suppl 3)(S3):1–10.

J.-H. Kim, I.-H. Kang, and K.-S. Choi. 2002. Unsupervised named entity classification models and their ensembles. In *Proceedings of COLING*.

M. Li, X. Chen, X. Li, B. Ma, and P.M.B. Vitanyi. 2004. The similarity metric. *IEEE Transactions on Information Theory*, 50(12):3250–3264.

W. Liao and S. Veeramachaneni. 2009. A simple semi-supervised algorithm for named entity recognition. In *Proceedings of the NAACL HLT Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 58–65.

F.D. Meulder and W. Daelemans. 2003. Memory-based named entity recognition using unannotated data. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL - Volume 4*, pages 208–211.

G. K. Palshikar. 2012. Techniques for named entity recognition: A survey. In *Collaboration and the Semantic Web: Social Networks, Knowledge Networks and Knowledge Resources*, pages 191–217. IGI Global.

S. Patil, S. Pawar, G. K. Palshikar, S. Bhat, and R. Srivastava. 2013. Unsupervised gazette creation using information distance. In *Proceedings of the 18th International Conference on Application of Natural Language to Information Systems (NLDB), LNCS 7934*. Springer-Verlag.

S. Pawar, R. Srivastava, and G. K. Palshikar. 2012. Automatic gazette creation for named entity recognition and application to resume processing. In *Proceedings of ACM COMPUTE*.

P.P. Talukdar, T. Brants, M. Liberman, and F. Pereira. 2006. A context pattern induction method for named entity extraction. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL)*, pages 141–148.

M. Thelen and E. Riloff. 2002. A bootstrapping method for learning seman-tic lexicons using extraction pattern contexts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Agriculture News Source websites. 2012. `cornandsoybeandigest.com`, `deltafarmpress.com`, `southwestfarmpress.com`, `southeastfarmpress.com`, `westernfarmpress.com`. [Online; accessed Mar. 2012].