GWU English TAC-KBP EL Diagnostic Task with Name Mention

Ayah Zirikly
Department of Computer Science
George Washington University
ayaz@gwu.edu

Yassine Benajiba
Luki Labs LLC
New York, NY
yassine@lukilabs.com

Mona Diab
Department of Computer Science
George Washington University
mtdiab@gwu.edu

Abstract

This paper describes the entity linking system participating in the 2015 Knowledge Base Population (KBP) track at the Text Analysis Conference (TAC) by GWU's Natural Language Processing (NLP) group (Care4Lang) in collaboration with the NLP consulting company Luki Labs. Our proposed system uses a supervised modeling approach with a feature set that targets the overlapping information between the query and the candidate entities from the KB. In addition, it uses an unsupervised approach to cluster the mentions that don't have a reference in the KB. It is a first participation for both teams and the attained results are promising and encouraging for further research.

1 Introduction

This paper describes our participation in the 2015 TAC-KBP track. Our goal was to specifically focus on the entity linking disambiguation aspect of the task, accordingly we did not participate in the mention identification and detection components of the task. Additionally, we only handle English name gold mentions.
Our system models the problem as a binary classification task for the candidate entities with respect to a reference Knowledge Base (KB) unique identifier. We use Support Vector Machines (SVMs) to generate a predictive model that uses a feature set reflecting the overlapping information of both entity profiles, namely:

1. the first profile belongs to the target query entity. The profile is created by co-referencing all the mentions of the entity, finding all the other Names Entities (NE) that are mentioned in the text. The collection of the entity's name variants mentioned in the text together with the other NE, gender information (where applicable); and,

2. the second profile belongs to the candidate entity. It is created using information from the KB. This second profile consists of the name of the entity, its aliases, and relations together with its description text. Thereafter, the overlapping information of both profiles is passed to a supervised classifier to build our linking model, and we cluster the entities for which we could not find any good candidates (NIL).

The obtained results are promising; notwithstanding the lower recall performance of 37.8% (due to our focus on the linking disambiguation), our approach yielded a precision of 77.5%. This constitutes a solid framework for our future (and more complete) system.

The remainder of the paper presents the related work (Section 2) to put our work in perspective. Thereafter, it describes our approach in Section 3 and the obtained results in Section 4. Section 5 discusses the results and Section 6 briefly reports our conclusions and provides a reflection on future work.

2 Related

Entity Linking and Disambiguation have been heavily studied in NLP due to its significant importance in many domains (e.g. biomedical, newswire, etc.). Since 2009, the TAC-KBP Entity Linking (EL) task aims to link a given named entity mention from a source document to an existing Knowledge Base (KB) entry (Ji et al., 2014). Additionally, EL requires clustering mentions that cannot be linked to an entry in the Knowledge Base to a NIL cluster. (Han and Sun, 2011) proposed a generative probabilisitic model that uses entity knowledge such as popularity, name and context knowledge for the EL task. (Zheng et al., 2010) proposed a framework for learning ranking to link entities with the KB. The authors use an extensive list of features (surface, contextual, and special features such as country names and city names). Then, they use a ranking perceptron to generate the ranking of candidates. Other ranking methodologies are proposed such as (Charton et al.,) that, following generating the candidates based on Wikipedia pages, they re-rank the links based on the mutual information between all the named entities in the document.

(Han and Zhao, 2010) introduce a knowledge-based method that captures and leverages the structural semantic knowledge in multiple knowledge sources (such as Wikipedia and WordNet) in order to improve the disambiguation performance.

Other approaches that focus on the aliasing side of the problem on parallel data without the use of external resources are proposed in (Zirikly and Diab, 2013). The authors experiment with a number of features (e.g. Co-occurrence Frequency, Relative Rank Order, etc.) and apply clustering on the resulting feature vectors using cosine similarity.

3 Approach

We target the English Entity Linking diagnostic task that takes as input the gold name mentions (mention type = NAM) and the offsets in the documents. For each queried mention we either provide the KB unique identifier (MID in Freebase) if the mention is linkable, otherwise NIL. For the latter, we cluster all the NIL instances into k clusters, where each cluster should represent all the different mention strings that map to the same entity (Figure 1 shows our resolution system's input and output framework).

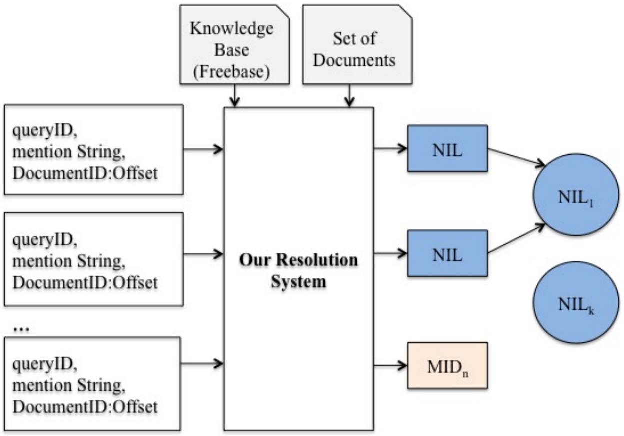
Knowledge Base (Freebase)
Set of Documents
queryID, mention String, DocumentID:Offset
queryID, mention String, DocumentID:Offset
...
queryID, mention String, DocumentID:Offset
Our Resolution System
NIL
NIL
MID$_n$
NIL$_1$
NIL$_k$

Figure 1: The GWU Resolution System

3.1 Preprocessing

There are several steps that need to be performed on the mention string (name normalization), the documents, and some textual preprocessing on the queries to the KB, then clustering of the the results (if NIL). The list of the preprocessing steps are:

1. **Name Normalization**. We normalize the surface form as it appears in the query (in case it is not an abbreviation) by applying the following: i) Inserting a space in case of capital letters in the middle of the input string (e.g. BarackObama =>Barack Obama); ii) Capitalizing the first character of an entity and enforcing that the rest of the characters are lowercased (e.g. $OBAMA \Rightarrow Obama$);

2. **Preprocessing on Documents** For all the documents that are listed in the query file, we perform: Tokenization (TOK), Part of Speech (POS) tagging, Named Entity Recognition (NER) (including entity types), and Coreference Resolution (CR);

3. **Preprocessing on KB**: For the textual information in the KB (e.g. description), we apply: TOK, NER, and CR;

   For the remainder of this paper, we assume tokenized documents and KB descriptions/textual information.

3.2 Entity Linking Approach

First, we query the KB with the surface form of the mention in case it represents an abbreviation. Otherwise, we query on the name normalized version of the surface form (*Norm*) and augment the querying terms with All-Caps($Cap_{N}orm$), where all the letters of the mention are in uppercase, and All-Small Norm ($Small_{N}orm$), where all the letters of the mention are in lowercase. Due to the large number of KB's unique identifiers returned (MID in Freebase), we limit the size of the list to 30 MIDs where the results from *Norm* have the highest priority followed by $Cap_{N}orm$, then lastly $Small_{N}orm$. If the list of results is empty, we mark the query as NIL and apply clustering on the set of mention strings that are not linkable (further details in 3.3). For every MID generated, we extract the following list of features:

Name Feature The surface form of the input query mention string (if abbreviation), and its normalized version with the augmented name variations.

Rank Feature For every query mention string, Freebase generates a list of results (MIDs). The results' ranking (relevance score) generated by Freebase is popularity-based, i.e. it is based on the count of inbound and outbound links in Freebase and Wikipedia. This feature adopts the natural default ranking of Freebase where the first MID gets a ranking = 1, and so on.

KB unique identifier The unique identifier of the KB (MID in Freebase), if querying KB is not NIL.

String Similarity Features We use two different similarity measurement features:

1. Relative Overlap with KB Mention (*RelOverlapSurf*): This feature captures the longest common substring between the surface/normalized query mention string (*queryStr*) and the surface form of the mention string in KB (*kbStr*). We normalize the size of overlap by $max\_length(queryStr, kbStr)$;
2. Maximum Overlap With Aliases (*MaxOverlapAlias*): This feature returns the maximum length of common substring between each of the input mention's aliases in the KB and the input mention string as shown in Equation 1

$$\begin{gathered} MaxOverlapAlias_j = Max_j(S_i) : \\ S_i = Max(substring(alias_{ji}, queryStr_j)), \\ i \in \{1, \ldots, \#aliases\}, j \in \{1, \ldots, \#queries\} \end{gathered} \tag{1}$$

Entity Type Features In our approach we use the following entity types: Person (PER), Location (LOC), Organization (ORG), Geopolitical entity (GPE), and Facility (FAC). We generate the entity type for both the textual data *Type-doc* and the KB data *Type-kb* as follows:

1. *Type-doc*: We use the Stanford CoreNLP (Manning et al., 2014) NER component;
2. *Type-kb*: We develop our own multiclass Support Vector Machine (SVM) classifier (Vapnik and Vapnik, 1998) that uses the following features: a) Normalized surface form of *queryStr*; b) The surface form of the mention as it appears in the KB; c) *Type-doc*; d) Set of inbound and outbound relations that are connected to the mention; e) Gazetteers (PER, LOC, ORG); f) Set of trigger words for FAC class (e.g. university). We added this feature due to the fact that Stanford NER cannot capture and provide the entity type for the mention in KB, where the mentions are connected via graphical relations.

Table 1, depicts the Precision (P), Recall (R), and F1-measure of our classifier on each of the five classes.

	P	R	F1-score
PER	92.5	100	96.1
LOC	86.4	69.5	77
ORG	96.3	90.1	93.1
GPE	98.3	93.4	95.8
FAC	79.4	52.9	63.5

Table 1: Entity Type Classifier Results

Gender Features We generate two features for gender, where the feature set values={Female, Male, Other, NA}. The first feature is based on information from the documentsGender-doc, while the second is based on KB *Gender-kb*:

1. *Gender-doc*: We assign the gender of the input mention using the textual information in the documents. Mainly, we rely on the coreference clusters where the mention gender is assigned Female if the pronoun *she* is also a member of the same coreference cluster. For instance if we have the following sentence: *Obama* is going to make his speech after *he* meets with Kerry. *Obama* and *he* both belong to the same coreference cluster, thus Obama gender is assigned the Male label;
2. *Gender-kb*: We assign the gender of the input mention using KB relations, specifically Female and Male co-occurrence relations.

Semantic Features These features involve the use of the description provided by Freebase and the text in the input mention document:

1. *Desc-doc*: This feature reflects the size of common words between the description (if exists) in Freebase, and the document. We normalize the text by the number of words in both texts. subtracting the number of common words;
2. *Desc-doc-NE*: This feature represents the number of common Named Entities between both texts.

3.3 NIL Clustering

For all the input mentions that were not linked to an entry in the KB, we perform clustering. First, for every mention we generate a feature vector that comprises the following features: 1) *Name Feature*; 2) *Type-doc*; 3)*Gender-doc*; 4) Coreference cluster ID concatenated with the name of the file in the form of *corefClusterID:fileName*; 5) POS tag; 6) sentence2vec: This feature uses the word vector representation of the words W in the sentence S and calculate their average, as shown in Equation 2. For the word vectors, we use word2vec (Mikolov et al., 2013) vectors generated from Google News with dimension=300. Then, we apply the K-Means clustering algorithm on the generated feature vectors where k (number of clusters) is selected empirically.

$$sentence2vec(S) = \Big(\sum_{i}^{\#words} W_{1i}, \ldots, \sum_{i}^{\#words} W_{di} \Big)$$

$$d = [1, dimension] \quad (2)$$

4 Experiments & Results

4.0.1 Experiment Setup & Tools

Knowledge Base setup We load the Freebase dump provided by the task coordinator (LDC2015E42) in a standalone Apache Jena Fuseki server[1] (SPARQL server) which we can query using SPARQL protocol over HTTP. In conjunction, we use TDB2 Loader for data storage that protects the dataset from corruption and add Apache Solr index. Then, we use SPARQL 1.1 for querying the data using Java Jena API.

Tools In our system we employ the following tools:

- Stanford CoreNLP (Manning et al., 2014): We use this toolkit for the following tasks: Tokenization, POS tagging, Coreference Resolution, Named Entity Recognition (NER);
- WEKA (Hall et al., 2009): We use WEKA for classification and clustering

Experiments We use the training data provided by the task for the year of 2015 (LDC2015E75). The training data consists of 30839 queries that includes Chinese (CMN), English (EN), and Spanish (ES) name (NAM) and nominal (NOM) mentions. In our experiments, we only target NAM English entries (12180 queries). The English source documents cover two genres: discussion forums *DF* (83 files) and newswire documents *NW* (85 documents). The filtered training queries genre-wise distribution is 7369 queries that are mentioned in *DF* documents, as opposed to 4811 queries in *NW* documents. The type/token ration (TTR) in the training and test data is $\approx 20\%$ in NW , and $\approx 15\%$ in DF.
Table 2 shows the same stats for the test data provided by the task (LDC2015E103).

In our training phase, we combine the DF and NW data in a single model since we have more DF data, hence the training models will be more

[1] https://jena.apache.org/documentation/serving˙data/

	NW	DF	Total
EN & NAM	6377	7664	14041
num_documents	83	85	168

Table 2: Test Data Stats

skewed. In future, we will experiment with genre separated models. We build a binary classifier using SVMs for both the DF and NW jointly. We use K-Means clustering for grouping NIL instances. We experiment with multiple number of clusters $k = 1000, 1500, 2500, 3500, 4000$. The selection of k was empirically based on the gold number of clusters in the training data.

Results Table 3 shows GWU team results for Precision, Recall, and F1-score for English when $k = 4000$. The measurements used are[2]:

- strong_link_match: micro-averaged evaluation of links. A system link must have the same KB gold link identifier to be counted correctly;
- strong_linked_mention_match: considers non-NIL mentions that are linked to KB identifier;
- strong_nil_match: micro-averaged evaluation of NIL mentions;
- strong_all_match: micro-averaged link evaluation of all mentions. A mention is counted as correct if is either a link or a NIL match;

The measurements used for clustering are[2]:

- pairwise: measures the proportion of mention pairs occurring in the same cluster in both gold and predicted clusterings;
- MUC: counts the number of edits required to translate the gold clustering into the prediction;
- B_cubed: assesses the proportion of each mention's cluster that is shared between gold and predicted clusterings;

Additionally, Table 4 depicts the results of our system in combination of entity type prediction. The measurements used are[2]:

- strong_typed_link_match: requires, in addition, correct entity type prediction to strong_link_match;
- strong_typed_nil_match: requires correct entity type prediction to strong_typed_nil_match;

measure	P	R	F1
strong_link_match	77.5	37.8	50.8
strong_linked_mention_match	95.9	46.7	62.8
strong_nil_match	36.5	73.8	48.8
strong_all_match	52	48	49.9
pairwise	90.4	48.2	62.9
MUC	74.7	67.8	71
B_cubed	72.4	53.2	61.3

Table 3: Precision (P), Recall (R), F-score (F1) of GWU team for English (without type prediction), k=4000

- strong_typed_all_match: a correct link must have the same span, entity type, and KB identifier as a gold link. A correct NIL must have the same span as a gold NIL.

measure	P	R	F1
strong_typed_link_match	68.1	33.2	44.6
strong_typed_nil_match	32.9	66.6	44.1
strong_typed_all_match	46.2	42.7	44.4

Table 4: Precision (P), Recall (R), F-score (F1) of GWU team for English (with type prediction)

5 Discussion

Observing the results of the *strong_link_match* measure in Table 3, we obtain 50.8% F1 score linking performance where the precision (77.5%) is much higher than recall (37.8%). These results align with the features proposed in our system that focus on maximizing precision (e.g. *Gender-doc, Desc-doc*) after generating the list of candidates based on normalizing the input query mention; as opposed to applying additional query augmentation techniques that will increase recall.

On the other hand, we did not provide other alternatives to increase recall besides the pronoun-coreference_representative replacements (which is prone to error). Possible methods to increase performance is to search with partial name (e.g. last name), or to perform some edit distance similarities on the input mention query.

On the other hand, we note that our NIL matching performance *strong_nil_match* of F1-score=48.8% reflects a low precision and high recall. The high recall is due to the fact that we map anything not link-

[2] https://github.com/wikilinks/neleval/wiki/Evaluation

	NW			DF		
measure	P	R	F1	P	R	F1
strong_link_match	75.7	33.5	46.5	78.9	20.2	32.1
strong_linked_mention_match	94.2	41.7	57.8	97.2	24.8	39.6
strong_nil_match	27	58.1	36.9	44.8	37.1	40.6
strong_all_match	44.4	40.2	42.2	58.2	25.6	35.6
strong_typed_link_match	65.6	29	40.3	70	17.9	28.5
strong_typed_nil_match	22	47.3	30	42.6	35.3	38.6
strong_typed_all_match	37.6	34	35.7	53.4	23.5	32.6
pairwise	86.2	42.6	57	91.9	6.8	12.7
MUC	77	64.7	70.3	76.2	34.1	47.1
B_cubed	74.6	50.1	60	76.4	21.8	33.9

Table 5: Precision (P), Recall (R), F-score (F1) of GWU team for English NW & DF, k=4000

able in KB to NIL. When we combine the entity type prediction to the previous measures we get an F1-score=44.6% for *strong_link_match* and F1=44.1% for *strong_nil_match*. Our best predicted entity types are GPE, ORG and PER (F1=67.4, 32.9, 26.6 respectively).

As illustrated in Table 5, we note that *NW* performance surpasses *DF* in all the measures for linking and NIL clustering. Additionally, we note that when entity type prediction is added, the results of *DF* slightly decreases when compared to its counterpart results without type prediction.

6 Conclusion & Future Work

In this paper, we tackle the TAC diagnostic KBP-EDL task with name gold input mentions. We propose a supervised approach to link to the representative KB reference ID using a set of features such as gender, word co-occurrences of textual data related to the input mention, rank.... For the NIL mentions (not linked to an entry in KB), we apply k-means clustering on a different set of features that include word embedding of the text data in the document in addition to some common features with linking such as gender. In our future work, we will consider splitting the data based on genre and compare our current approach with genre-dependent models performance. We would also like to incorporate more features that will specifically improve recall.

Acknowledgment

This work was funded by the DARPA DEFT project with a subcontract from Columbia University.

References

Eric Charton, Marie-Jean Meurs, Ludovic Jean-Louis, and Michel Gagnon. Mutual disambiguation for entity linking.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.

Xianpei Han and Le Sun. 2011. A generative entity-mention model for linking entities with knowledge base. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 945–954. Association for Computational Linguistics.

Xianpei Han and Jun Zhao. 2010. Structural semantic relatedness: a knowledge-based method to named entity disambiguation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 50–59. Association for Computational Linguistics.

Heng Ji, HT Dang, J Nothman, and B Hachey. 2014. Overview of tac-kbp2014 entity discovery and linking tasks. In *Proc. Text Analysis Conference (TAC2014)*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Vladimir Naumovich Vapnik and Vlamimir Vapnik. 1998. *Statistical learning theory*, volume 1. Wiley New York.

Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. 2010. Learning to link entities with knowledge base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–491. Association for Computational Linguistics.

Ayah Zirikly and Mona Diab. 2013. Anear: Automatic named entity aliasing resolution. In *Natural Language Processing and Information Systems*, pages 213–224. Springer.