Entity Linking Korean Text: An Unsupervised Learning Approach using Semantic Relations

Youngsik Kim
KAIST / Korea, The Republic of
twilight@kaist.ac.kr

Key-Sun Choi
KAIST / Korea, The Republic of
kschoi@kaist.edu

Abstract

Although entity linking is a widely researched topic, the same cannot be said for entity linking geared for languages other than English. Several limitations including syntactic features and the relative lack of resources prevent typical approaches to entity linking to be used as effectively for other languages in general. We describe an entity linking system that leverage semantic relations between entities within an existing knowledge base to learn and perform entity linking using a minimal environment consisting of a part-of-speech tagger. We measure the performance of our system against Korean Wikipedia abstract snippets, using the Korean DBpedia knowledge base for training. Based on these results, we argue both the feasibility of our system and the possibility of extending to other domains and languages in general.

1 Introduction

A crucial step in creating the Web of Data is the process of extracting structured data, or RDF (Adida et al., 2012) from unstructured text. This step enables machines to read and understand unstructured Web pages that consist the majority of the Web. Three tasks play a part in extracting RDF from unstructured text: Named entity recognition(NER), where strings representing named entities are extracted from the given text; entity linking(EL), where each named entity recognized from NER is mapped to a appropriate resource from a knowledge base; and relation extraction(Usbeck et al., 2014). Although entity linking is an extensively researched field, most research done is aimed primarily for the English language. Research about entity linking for languages other than English have also been performed (Jakob et al., 2013), but most state-of-art entity linking systems are not fully language independent.

The reason for this is two-fold. Firstly, most entity linking systems depend on an existing system to perform named entity recognition beforehand. For instance, the system proposed by Usbeck (2014) uses FOX (Speck et al., 2014) to perform named entity recognition as the starting point. The problem with this approach is that an existing named entity recognition system is required, and thus the performance of entity linking is bound by the performance of the named entity recognition system. Named entity recognition systems for English achieve high performance even by utilizing a simple dictionary-based approach augmented by part-of-speech annotations, but this approach does not work in all languages in general. CJK[1] languages in particular are difficult to perform named entity recognition on because language traits such as capitalization and strict token separation by white-space do not exist. Other approaches to named entity recognition such as statistical models or projection models respectively require a large amount of annotated training data and an extensive parallel corpus to work, but the cost of creating these resources is also non-trivial. Some approaches to entity linking utilizing supervised and semi-supervised learning also suffer from the lack of manually annotated training data for some languages. Currently, there is no proper golden standard dataset for entity linking for the Korean language.

In this paper, we present an entity linking system for Korean that overcomes these obstacles with an unsupervised learning approach utilizing semantic relations between entities obtained from a given knowledge base. Our system uses these semantic relations as hints to learn feature values

[1] Chinese, Japanese, Korean

Proceedings of the 19th Conference on Computational Language Learning, pages 132–141,
Beijing, China, July 30-31, 2015.

for both named entity recognition and entity linking. In Section 3, we describe the requirements of our system, and present the architecture of both the training and actual entity linking processes. In Section 4, we compare the performance of our system against both a rule-based baseline and the current state-of-art system based on Kim's (2014) research.

2 Related Work

The current state-of-art entity linking system for Korean handles both named entity recognition and entity linking as two separate steps (Kim et al., 2014). This system uses hyperlinks within the Korean Wikipedia and a small amount of text manually annotated with entity information as training data. It employs a SVM model trained with character-based features to perform named entity recognition, and uses the TF*ICF (Mendes et al., 2011) and LDA metrics to disambiguate between entity resources during entity linking. Kim (2014) reports an F1-score of 75.66% for a simplified task in which only surface forms and entities which appear as at least one hyperlink within the Korean Wikipedia are recognized as potential entity candidates.

Our system utilizes relations between entities, which can be said to be a graph-based approach to entity linking. There have been recent research about entity linking that exploit the graph structure of both the named entities within text and RDF knowledge bases. Han (2011) uses a graph-based collective method which can model and exploit the global interdependence between different entity linking decisions. Alhelbawy (2014) uses graph ranking combined with clique partitioning. Moro (2014) introduces Babelfy, a unified graph-based approach to entity linking and word sense disambiguation based on a loose identification of candidate meanings coupled with a densest subgraph heuristic which selects high-coherence semantic interpretations. Usbeck (2014) combines the Hypertext-Induced Topic Search (HITS) algorithm with label expansion strategies and string similarity measures.

There also has been research about named entity recognition for Korean. Kim (2012) proposes a method to automatically label multilingual data with named entity tags, combining Wikipedia meta-data with information obtained through English-foreign language parallel Wikipedia sentences. We do not use this approach in our system because our scope of entities is wider than the named entity scope defined in the MUC-7 annotation guidelines, which is the scope of Kim's research.

3 The System

Due to the limitations of performing entity linking for the Korean language described in Section 1, our system is designed with some requirements in mind. The requirements are:

- The system should be able to be trained and ran within a minimal environment, which we define as an existing RDF knowledge base containing semantic relations between entities, and a part-of-speech tagger. In this paper, we use the 2014 Korean DBpedia RDF knowledge base and the ETRI Korean part-of-speech tagger.
- The system should be able to perform entity linking without using external information not derived from the knowledge base.

We define the task of our system as follows: Given a list of entity uniform resource identifiers(URI) derived from the knowledge base, annotate any given text with the appropriate entity URIs and their positions within the text.

3.1 Preprocessing

The preprocessing step of our system consists of querying the knowledge base to build a dictionary of entity URIs and their respective surface forms. As we are using the Korean DBpedia as our knowledge base, we define all resources with a URI starting with the namespace 'http://ko.dbpedia.org/resource/' and which are not mapped to disambiguation nor redirection Wikipedia pages as valid entities. For each entity, we define all literal strings connected via the property 'rdfs:label' to the entity and all entities that disambiguate or redirect to the entity as possible surface forms.

The dictionary that results from preprocessing contains 303,779 distinct entity URIs and 569,908 entity URI-surface form pairs.

3.2 Training

After the preprocessing step, our system performs training by adjusting feature values based on data from a large amount of unannotated text documents. As the given text is not annotated with entity data, we use the following assumption to help distinguish potential entities within the text:

Assumption. Entity candidates which have a high degree of semantic relations with nearby entity candidates are likely actual entities. We define an ‘entity candidate’ of a document as a substring-entity URI pair in which the substring appears at a specific position within the document and the pair exists in the dictionary created during preprocessing, and a ‘semantic relation’ between two entity candidates c_1, c_2 ($RelPred(c_1, c_2)$) as an undirected relation consisting of all predicates in the knowledge base that connect the entity URIs of the entity candidates.

The basis for this assumption is that some mentions of ‘popular’(having a high degree within the knowledge base RDF graph) entity URIs will be accompanied with related terms, and that the knowledge base will have RDF triples connecting the entity to the other entity URIs representing the related terms. Thus we assume that by selecting all entity candidates with a high degree of semantic relations, these candidates display features more representative of actual entities than the remaining entity candidates do.

For each document in the training set, we first perform part-of-speech tagging to split the text into individual morphemes. Because the concatenation of the morphemes of a Korean word is not always identical to the word itself, we transform the morphemes into ‘atomic’ sub-strings which represent minimal building blocks for entity candidates and can have multiple POS tags.

After part-of-speech tagging is complete, we then gather a set of entity candidates from the document. We first find all non-overlapping substrings within the document that correspond to entity surface forms. It is possible for these sub-strings to overlap with each other([[Chicago] Bulls]); and because the average length of entity surface forms in Korean is very short(between 2 to 3 characters), we opt to reduce the problem size of the training process by choosing only the substring with the longest length when multiple substrings overlap with each other.

As to further reduce the number of entity candidates we consider for training, we only use the entity candidate with the most ‘popular’ entity URI per group of candidates that share the same substring within the document. We define the ‘popularity’ of an entity URI in terms of the RDF triples in the knowledge base that has the URI as the *object*. More formally, we define $UriPop(c)$ for an entity candidate c with the entity URI c_u with the equations below. A larger $UriPop$ value means a more ‘popular’ entity URI, and $Pred$ is meant to prevent overly frequent predicates in the knowledge base from dominating $UriPop$.

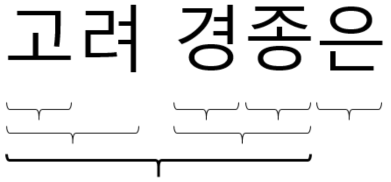

Figure 1: All possible entity candidates within the text fragment ‘Gyeongjong of Goryeo is’ (Kim et al., 2014)

$$UriPop(c) = log(\sum_{p \in KBpredicates} Pred(p) \times |\{s|(s, p, c_u) \in KB\}|) \quad (1)$$

$$Pred(p) = 1 - \frac{|\{(s, o)|(s, p, o) \in KB\}|}{|\{(s, p, o) \in KB\}|} \quad (2)$$

At this stage of the training process, we have a set of non-overlapping entity candidates. We now classify these candidates into two classes: e_T(entity) and e_F(non-entity) according to our previous assumption. We measure the degree of semantic relations of an entity candidate c, $SemRel(c)$, with the following equation where N_c is the set of entity candidates which are within 20 words from the target candidate in the document:

$$SemRel(c) = \frac{\sum_{c' \in N_c} \sum_{p \in RelPred(c,c')} Pred(p)}{|N_c|} \quad (3)$$

Since we do not have enough evidence at this point to distinguish entities from non-entities, we use semantic relations from all nearby entity candidates to calculate $SemRel$. We order the entity candidates into a list in decreasing order of $SemRel$, and classify entity candidates from the start of the list into e_T until either 1) $\frac{e_T}{\text{document word \#}} > l_w$ or 2) $SemRel(c) < l_s$ are satisfied, where both l_w and l_s are constants that continuously get adjusted during the training process. The remaining entity candidates all get classified into e_F.

We now update the current feature values based on the entity candidates in e_T and e_F. For each feature f, we first define two sub-classes $e_{fT} = \{e|e \text{ has } f \wedge e \in e_T\}$ and $e_{fF} = \{e|e \text{ has } f \wedge e \in e_F\}$. We then update the feature value of f from $\frac{\alpha}{\beta}$ to $\frac{\alpha+|e_{fT}|}{\beta+|e_{fT}|+|e_{fF}|}$, which represents the probability of an entity candidate having feature f to be classified into e_T(is an entity). The full list of features we used is shown below:

f_1**: String length** The length (in characters) of the sub-string of the candidate.

f_2**: POS tag** The POS tag(s) of the sub-string of the candidate. If multiple POS tags exist, we take the average of the f_2 values for each tag as the representive f_2 value.

f_3**: Head POS tag** The first POS tag of the sub-string of the candidate.

f_4**: Tail POS tag** The last POS tag of the sub-string of the candidate.

f_5**: Previous POS tag** The POS tag of the sub-string right before the candidate. If the candidate is preceded by white-space, we use the special tag 'BLANK'.

f_6**: Next POS tag** The POS tag of the sub-string right after the candidate. If the candidate is followed by white-space, we use the special tag 'BLANK'.

f_7**: UriPop** The $UriPop$ score of the entity URI of the candidate. Since $UriPop$ has a continuous range, we keep separate features for $UriPop$ score intervals of 0.2.

Given these features, we define $IndScore(c)$ of an entity candidate c, which represents the overall probability c would be classified in e_T independently of its surrounding context, as the average of the feature values for c.

We also define $WeightedSemRel(c)$ as the amount of evidence via semantic relations c has of being classified in e_T. We define this score in terms of semantic relations relative to the $UriPop$ score of c in order to positively consider entity candidates which have more semantic relations than their entity URIs would normally have.

Finally, we define $EntityScore(c)$ representing the overall evidence for c to be in e_T. The respective equations for these scores are shown below.

$$WeightedSemRel(c) = \frac{\sum_{c' \in N_c} \sum_{p \in RelPred(c,c')} Pred(p) \times UriPop(c')}{UriPop(c)} \tag{4}$$

$$EntityScore(c) = IndScore(c) + WeightedSemRel(c) \tag{5}$$

As we want to assign stronger evidence for semantic relations with entity candidates that are likely actual entities (as opposed to relations with non-entities), we define $WeightedSemRel$ to be have a recursive relation with $EntityScore$.

We end the training process for a single document by computing the $EntityScore$ for each entity candidate in e_T and e_F, and adding these scores respectively into the lists of scores $dist_T$ and $dist_F$. As the $EntityScore$ for the same entity candidate will change as training proceeds, we only maintain the 10,000 most recent scores for both lists.

3.3 Entity Linking

Since the actual entity linking process is also about selecting entity candidates from the given document, the process itself is similar to the training process. We list the differences of the entity linking process compared to training below.

When we choose the initial entity candidates, we do not remove candidates with overlapping sub-strings. Although we do limit the number of candidates that map to the same sub-string, we choose the top 3 candidates based on $UriPop$ instead of just 1.

We compute the $EntityScore$ for each entity candidate without performing candidate classification nor feature value updates. Although the value of $EntityScore(c)$ is intended to be proportional to the possibility the candidate c is actually an entity, we need a way to define a threshold constant to determine which candidates to actually classify as entities. Thus, we then normalize any given $EntityScore(c)$ score of an entity candidate c into a confidence score $Conf(c)$, which represents the relative probability of c being a member of e_T against being a member of e_F. This is computed by comparing the score against the lists $dist_T$ and $dist_F$, as shown in the following equations:

$$ConfT(c) = \frac{|\{x|x \in dist_T \wedge x < EntityScore(c)\}|}{|dist_T|} \tag{6}$$

$$ConfF(c) = \frac{|\{x | x \in dist_F \wedge x > EntityScore(c)\}|}{|dist_F|} \tag{7}$$

$$Conf(c) = \frac{ConfT(c)}{ConfT(c) + ConfF(c)} \tag{8}$$

$Conf$ is a normalized score which satisfies $0 \leq Conf(c) \leq 1$ for any entity candidate c. This gives us the flexibility to define a threshold $0 \leq \gamma \leq 1$ so that only entity candidates satisfying $Conf(c) \geq \gamma$ are classified as entities.

Algorithm 1 The entity selection algorithm

```
$E \leftarrow []$
for all $c \in C$ do
  if $Conf(c) \geq \gamma$ then
    $unsetE \leftarrow []$
    $valid$ = true
    for all $e \in E$ do
      if sub-string of $c$ contains $e$ then
        $unsetE \leftarrow unsetE + e$
      else if sub-string of $c$ overlaps with $e$ then
        $valid$ = false
      end if
    end for
    if $valid$ is true then
      $E \leftarrow E + c$
      for all $e \in unsetE$ do
        $E \leftarrow E - e$
      end for
    end if
  end if
end for
return $E$
```

Algorithm 1 shows the entity selection process, where C is initialized as a list of all entity candidates ordered by decreasing score of $Conf$. We only select entity candidates which have a confidence score of at least γ. When the sub-strings of multiple entity candidates overlap with each other, we prioritize the candidate with the highest confidence with the exception of candidates that contain(one is completely covered by the other, without the two being identical) other candidates in which we choose the candidate that contains the other candidates.

4 Experiments

4.1 Entity Linking for Korean

4.1.1 The Dataset

Although the dataset used by Kim (2014) exists, we do not use this dataset because it is for a simplified version of the entity linking problem as discussed in Section 2. We instead have created a new dataset intended to serve as answer data for the entity linking for Korean task, based on guidelines that were derived from the TAC KBP 2014[2] guidelines. The guidelines used to annotate text for our dataset are shown below:

- All entities must be tagged with an entity URI that exists in the 2014 Korean DBpedia knowledge base.
- All entities must be tagged with an entity URI which is correct to appear within the context of the entity.
- All entity URIs must identify a single thing, not a group of things.
- Verbs and adjectives that have an equivalent noun that is identified by an entity URI should not be tagged as entities.
- Only words that directly identify the entity should be tagged.
- If several possible entities are contained within each other, the entity that contains all other entities should be tagged.
- If several consecutive words each can be represented by the same entity URI, these words must be tagged as a single entity, as opposed to tagging each separate word.
- Indirect mentions of entity URIs (ex: pronouns) should not be tagged even if coresolution can be performed in order to identify the entity URI the mention stands for.

Our dataset consists of 60 Korean Wikipedia abstract documents annotated by 3 different annotators.

[2] http://nlp.cs.rpi.edu/kbp/2014/elquery.pdf

4.1.2 Evaluation

We evaluate our system against the work of Kim (2014) in terms of precision, recall, and F1-score metrics. As Kim's (2014) system(which we will refer to as KEL2014) was originally trained with the Korean Wikipedia, it required only slight adjustments to properly operate for our dataset.

We first train our system using 4,000 documents randomly gathered from the Korean Wikipedia. We then evaluate our system with our dataset, while adjusting the confidence threshold γ from 0.01 to 0.99 in increments of 0.01. We compare our results with those of the best-performing settings of KEL2014.

4.1.3 Results

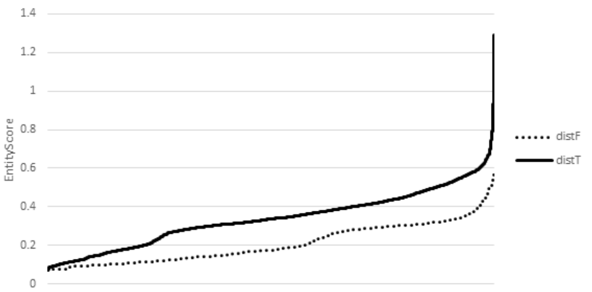

Figure 2: The distribution of $dist_T$ and $dist_F$ after training with 4,000 documents

The results of the training process is shown in Figure 2. We can see that the $EntityScore$ values of entity candidates classified in e_T are generally higher than those in e_F. This can be seen as evidence to our claim that the features of entities with a high degree of semantic relations can be used to distinguish entities from non-entities in general.

We show additional evidence to this claim with Table 1, which lists the feature values for the feature f_1(sub-string length) after the training process is complete. We observe that this feature gives relatively little weight to entity candidates with a substring of length 1, which is consistent with our initial observation that most of these candidates are not entities.

Length	e_T	$e_T + e_F$	Value
1	37629	312543	0.120
2	41355	168672	0.245
3	20164	41098	0.490
4	12542	19619	0.637
5	13953	19722	0.704
6	5299	7503	0.698
7	3223	4187	0.753
8	2686	4282	0.616
9	2346	3034	0.751
10	922	1099	0.774
11	917	1011	0.830
12	477	546	0.750
13	249	276	0.687
14	218	274	0.615
15	155	170	0.618
16	109	116	0.567
17	99	118	0.523
18	48	52	0.434
19	49	55	0.433
20	34	40	0.384

Table 1: Feature values of the feature f_1 after training with 4,000 documents

Figure 3 shows the performance of our system compared to the performance of KEL2014 against our dataset. As we increase the confidence threshold, the recall decreases and the precision increases. The maximum F1-score of our system using our dataset, 0.630, was obtained with the confidence threshold $\gamma = 0.79$. This is an improvement over KEL2014 which scored an F1-score of 0.598. Although the performance difference itself is small, this shows that our system trained with unannotated text documents performs better than KEL2014, which is trained with both partially annotated documents (Wikipedia articles) and a small number of documents completely annotated with entity data. This also shows that KEL2014 does not perform as well outside the scope of the simplified version of entity linking it was designed for.

Our system currently is implemented as a simple RESTful service where both training and actual entity linking are initiated via POST requests. Our system can be deployed to a server at its initial state or at a pre-trained state.

4.2 Alternative Methods and Deviations

4.2.1 Using Feature Subsets

In order to test the effectiveness of each feature we use in training, we compare the results obtained in Section 4.1.3 against the performance of our system trained with smaller subsets of the features shown in Section 3.2. Using the same training data as in Section 4.1.2, we evaluate the performance of our system using two different subsets of features as shown below:

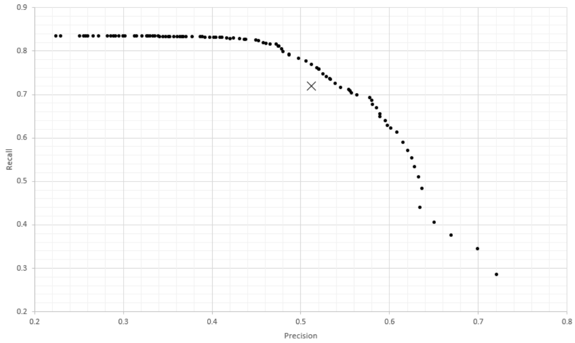

Figure 3: The performance of our system against KEL2014. The dots represent the results of our system, and the cross represents the results of KEL2014.

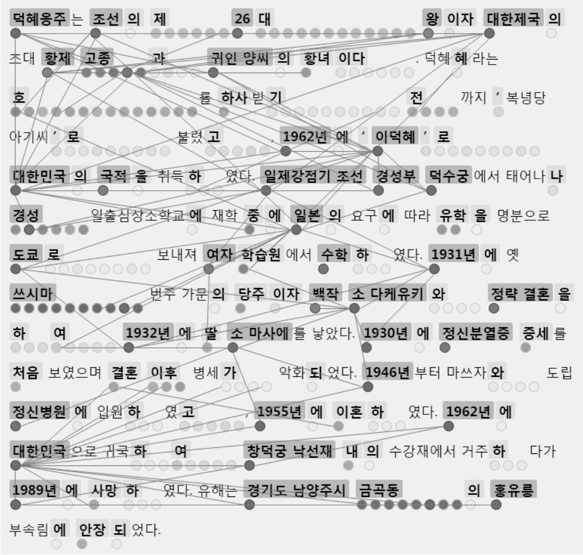
덕혜옹주 는 조선 의 제 26 대 왕 이자 대한제국 의
조대 황제 고종 과 귀인 양씨 의 황녀 이다 . 덕혜 혜 라는
호 를 하사 받 기 전 까지 ' 복녕당
아기씨 ' 로 불렸 고 , 1962년 에 ' 이덕혜 ' 로
대한민국 의 국적 을 취득 하 였다. 일제강점기 조선 경성부 덕수궁 에서 태어나 나
경성 일출심상소학교 에 재학 중 에 일본 의 요구 에 따라 유학 을 명분으로
도쿄 로 보내져 여자 학습원 에서 수학 하 였다. 1931년 에 옛
쓰시마 번주 가문 의 당주 이자 백작 소 다케유키 와 정략 결혼 을
하 여 1932년 에 딸 소 마사에 를 낳았다. 1930년 에 정신분열증 증세 를
처음 보였으며 결혼 이후 병세 가 악화 되 었다. 1946년 부터 마쓰자 와 도립
정신병원 에 입원 하 였 고 , 1955년 에 이혼 하 였다. 1962년 에
대한민국 으로 귀국 하 여 창덕궁 낙선재 내 의 수강재에서 거주 하 다가
1989년 에 사망 하 였다. 유해는 경기도 남양주시 금곡동 의 홍유릉
부속림 에 안장 되 었다.

Figure 4: A temporary web interface for our system showing entity linking results for a sample text snippet

Dev1 All features except POS-related ones: f_1, f_7.

Dev2 All POS-related features only: f_2, f_3, f_4, f_5, f_6.

Figures 5 and 6 shows the performance of our system using the feature subsets Dev1 and Dev2. For the feature subset Dev1, our system displays a maximum F1-score of 0.433 with $\gamma = 0.99$; for the features subset Dev2, our system performs best with $\gamma = 0.98$ for an F1-score of 0.568.

As expected, both deviations perform worse than our system trained with the full set of features. We observe that the performance of Dev1 is significantly worse than that of Dev2. This suggests that POS-based features are more important in distinguishing entities from non-entities than the other features.

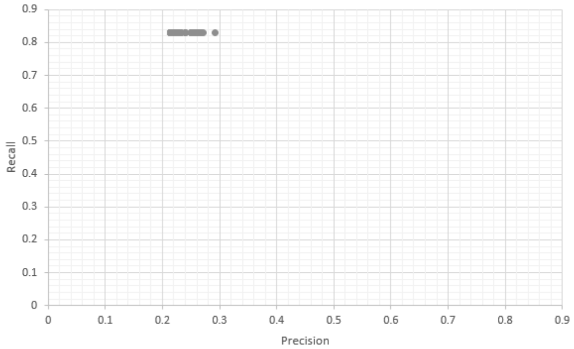

Figure 5: The performance of our system using the feature subset Dev1.

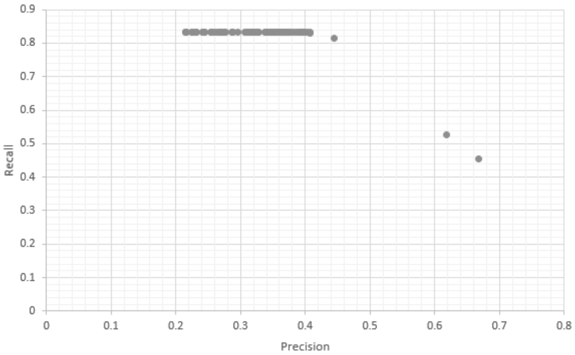

Figure 6: The performance of our system using the feature subset Dev2.

4.2.2 Using SVM Models

Kim (2014) effectively utilized trained SVM models to detect entities within text. Based on Kim's experiments, we investigate whether SVM can be also used to raise performance of our system.

Although we now need to base entity predictions with a trained SVM model instead of the confidence metric $Conf$, the training process described in Section 3.2 is mostly unaltered because it the classification of entity candidate into e_T and e_F results in the training data required to train a SVM binary classification model. The differences in the training process are shown below:

- After we process each document, a list of features is produced for each entity candidate classified as e_T or e_F. Instead of appending the $EntityScore$ values of these entity candidates to $dist_T$ and $dist_F$, we append the feature lists of each entity candidate themselves into two lists of lists, $list_T$ and $list_F$. These lists still contain only the data of the newest

10,000 entity candidates.

- After the entire training set is processed, we use the list of feature lists in $list_T$ and $list_F$ to train the SVM model. As the original features are not all numbers, we transform each list of features into 7-dimensional vectors by replacing each feature key into the feature value of the respective feature.

We use the same training data as in Section 4.1.2, and train a SVM model with a 3-degree polynomial kernel. We choose this kernel according to Kim's (2014) work, where Kim compares the performance of SVM for the task of entity linking for Korean using multiple kernels.

Our system, using a trained SVM model results in a F1-score of 0.569, which is about 0.07 lower than the best performance of our system using the confidence model. One possible reason our system performed worse using SVM classification is that the training data that we feed to the classifier is not correctly classified, but rather based on the semantic relations assumption in Section 3.2. As this assumption does not cover any characteristics of non-entities, the effectiveness of SVM decreases as many entity candidates which are actual entities get classified as e_F due to them not having enough semantic relations.

4.3 Application on Other Languages

As our system does not explicitly exploit any characteristics of the Korean language, it theoretically is a language-independent entity linking system. We investigate this point using Japanese as a sample language, and MeCab[3] as the part-of-speech tagger.

As no proper dataset for the entity linking for Korean task exists, we have created a new dataset in order to measure the performance of our system. As we believe many other languages will also lack a proper dataset, we must devise an alternative method to measure the performance of our system for other languages in general.

We use Wikipedia documents and the links annotated within them as the dataset to use for measuring performance of our system for languages other than Korean. Although the manually annotated links within Wikipedia documents do represent actual entities and their surface forms, we cannot completely rely on these links as answer data because of the following reasons:

- The majority of entities within a Wikipedia document are not actually tagged as links. This includes self-references (entities that appear within the document describing that entity), frequently appearing entities that were tagged as links for their first few occurrences but were not tagged afterwards, and entities that were not considered important enough to tag as links by the annotators. Due to the existence of so many untagged entities, we effectively cannot use precision as a performance measure.

- Some links represent entities that are outside the scope of our research. This includes links that point to non-existent Wikipedia pages ('red links'), and links that require coresolution to resolve.

Due to these problems, we only use the recall metric to measure the approximate performance of our system when using Wikipedia documents as answer data. We compare the performance of our system for Korean and Japanese by first training our system with 3,000 Wikipedia docuements, and measuring the recall of our system for 100 Wikipedia documents for both respective languages while adjusting the confidence threshold γ from 0.01 to 0.99 in increments of 0.01.

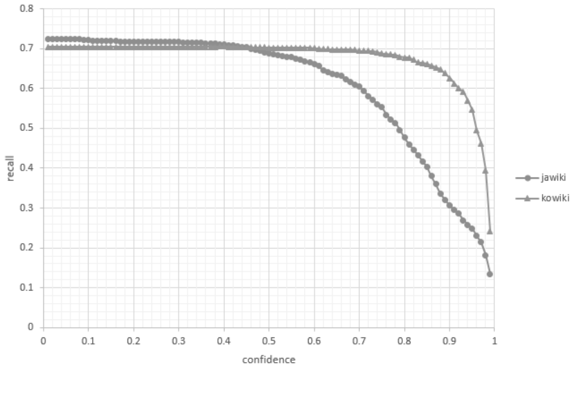

Figure 7: The recall of our system against Korean and Japanese Wikipedia documents

Figure 7 shows the recall of our system against Korean and Japanese Wikipedia documents. For the confidence threshold optimized via the experiments performed in Section 4.1 ($\gamma = 0.79$), our system shows a large difference of recall between

[3] http://taku910.github.io/mecab/

Korean and Japanese. Although this does not accurately represent the actual performance of our system, we leave the task of improving our system to display consistent performance across multiple languages as future work.

5 Future Work

As our system currently only uses labels of entity URIs to determine surface forms of entities, it can not detect entities with irregular surface forms. For instance, the entity ‘Steve˙Jobs’ has the labels ‘Steve Jobs’ and ‘Jobs’ within the Korean DBpedia knowledge base, but does not have the label ‘Steve’. This results in the inability of our system to detect certain entities within our dataset, regardless of training. We plan to improve our system to handle derivative surface forms such as ‘Steve’ for ‘Steve˙Jobs’, without relying on external dictionaries if possible.

Kim (2014) shows that a SVM classifier using character-based features trained with Wikipedia articles achieves better named entity recognition performance than a rule-based classifier using part-of-speech tags. Although our system uses trained features based on part-of-speech tags rather than a rule-based method, we may be able to remove even the part-of-speech tagger from the requirements of our system by substituting these features with the character-based features suggested in Kim’s (2014) work.

Finally, future work must be performed about replacing the part-of-speech tagger currently used in our system with a chunking algorithm that does not utilize supervised training. Our system currently utilizes the list of morphemes and their respective POS tags that are produced from the part-of-speech tagger. Since our system does not require this information to be completely accurate, a dictionary-based approach to chunking might be applicable as well.

6 Conclusion

In this paper, we present an entity linking system for Korean that utilizes several features trained with plain text documents. By taking an unsupervised learning approach, our system is able to perform entity linking with a minimal environment consisting of an RDF knowledge base and a part-of-speech tagger. We compare the performance of our system against the state-of-art system KEL2014, and show that our system outperforms KEL2014 in terms of F1-score. We also briefly describe variations to our system training process, such as using feature subsets and utilizing SVM models instead of our confidence metric.

Many languages including Korean are not as rich in resources as the English language, and the lack of resources might prohibit the state-of-art systems for entity linking for English from performing as well in other languages. By utilizing a minimal amount of resources, our system may provide a firm starting point for research about entity linking for these languages.

7 Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. R0101-15-0054, WiseKB: Big data based self-evolving knowledge base and reasoning platform)

References

B. Adida, I. Herman, M. Sporny, and M. Birbeck. RDFa 1.1 Primer. Technical report, World Wide Web Consortium, http://www.w3.org/TR/2012/NOTE-rdfa-primer- 20120607/, June 2012.

Usbeck, R., Ngomo, A. C. N., Rder, M., Gerber, D., Coelho, S. A., Auer, S., and Both, A. (2014). AGDISTIS-graph-based disambiguation of named entities using linked data. In The Semantic WebISWC 2014 (pp. 457-471). Springer International Publishing.

Daiber, J., Jakob, M., Hokamp, C., and Mendes, P. N. (2013, September). Improving efficiency and accuracy in multilingual entity extraction. In Proceedings of the 9th International Conference on Semantic Systems (pp. 121-124). ACM.

Speck, R., and Ngomo, A. C. N. (2014). Ensemble learning for named entity recognition. In The Semantic WebISWC 2014 (pp. 519-534). Springer International Publishing.

Y. Kim, Y. Hamn, J. Kim, D. Hwang, and K.-S. Choi, A Non-morphological Approach for DBpedia URI Spotting within Korean Text, Proc. of HCLT 2014, Chuncheon, 2014.

P. N. Mendes, M. Jakob, A. Garca-Silve, and C. Bizer, ”DBpedia spotlight: shedding light on the web of documents,” Proc. of i-SEMANTICS 2011 (7th Int. Conf. on Semantic Systems, 2011.

Han, X., Sun, L., and Zhao, J. (2011, July). Collective entity linking in web text: a graph-based method. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (pp. 765-774). ACM.

Alhelbawy, Ayman, and Robert Gaizauskas. ”Collective Named Entity Disambiguation using Graph Ranking and Clique Partitioning Approaches.”

Moro, Andrea, Alessandro Raganato, and Roberto Navigli. ”Entity linking meets word sense disambiguation: a unified approach.” Transactions of the Association for Computational Linguistics 2 (2014): 231-244.

Kim, S., Toutanova, K., and Yu, H. (2012, July). Multilingual named entity recognition using parallel data and metadata from wikipedia. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1 (pp. 694-702). Association for Computational Linguistics.