RPI_BLENDER TAC-KBP2014 Knowledge Base Population System

Yu Hong[1,2], Xiaobin Wang[1], Yadong Chen[1], Jian Wang[1], Tongtao Zhang[2], Jin Zheng[2], Dian Yu[2], Qi Li[2], Boliang Zhang[2], Han Wang[2], Xiaoman Pan[2], Heng Ji[2]

[1] Computer Science Department, Soochow University
[2] Computer Science Department, Rensselaer Polytechnic Institute
tianxianer@gmail.com, jih@rpi.edu

1 Introduction

This year the RPI_BLENDER team participated in the following four tasks: English Entity Discovery and Linking (Ji et al., 2014), Regular Slot Filling (Surdeanu and Ji, 2014), Slot Filling Validation and KBP Event Argument Extraction. The core algorithm of our Entity Discovery and Linking is described in (Zheng et al., 2014). Our Slot Filling and Slot Filling Validation are based on a Multi-dimensional Truth Finding Model (Yu et al., 2014). Our KBP Event Argument Extraction system is based on joint modeling (Li et al., 2013; Li et al., 2014; Li and Ji, 2014). So in this paper we mainly focus on the unpublished new work for each task.

2 Entity Discovery and Linking

Detecting and linking concept mentions in a textual document to an existing knowledge base is an important problem and has attracted attentions from the research communities in recent years. It provides disambiguation of concepts and thus allows humans and machines to better analyze the meaning of the document, extract and integrate new knowledge from the document with an existing knowledge base. However, this problem is very challenging due to name ambiguity, name inconsistency, and the lack of a fully integrated domain knowledge base. Most of the previous approaches exploited supervised or semi-supervised learning approaches which heavily relied on manually annotated training data. In this paper, we present a novel unsupervised collective inference and validation approach to disambiguate concept mentions and link them to entities in the knowledge base (KB). The approach leverages the rich semantic information of DBpedia entities for similarity computation and ranking validation.

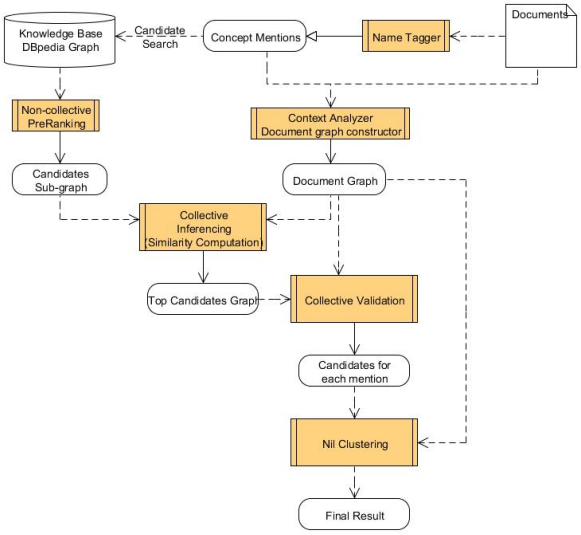

Figure 1: Approach Overview

Our Entity Discovery and Linking system contains four main components: 1) Name tagger, which aims to identify and classify entity mentions. The tagger combines heuristic rules with and a Linear-chain CRFs model. 2) Entity mention context analyzer to construct entity mention graph by analyzing the source document. The analyzer leverages name tagging, dependency parsing, relation extraction and event extraction to extract related context information. 3) Entity Linker that fetches candidate entities and link the entity mention to the

referent entity in the knowledge base (KB). This system exploitsrich semantic information of each entity from both source document and DBpedia to obtain final entity. 4) Nil Clustering which computes mention similarity score by comparing mention strings and context similarity score based on name type and contextual information.

2.1 Name tagging

Our baseline name tagger (Li et al., 2012a) is based on linear-chain CRFs model trained from ACE 2003-2005 corpora. We made the following improvement.

- To improve the coherency of the tagging results from the same document, we do a majority voting of all coreferential mentions and propagate a consistent label for each entity in a document.
- Gazetteer is an important type of feature for name tagging. The training data of our name tagging is from news articles published before 2005. Therefore, there exists a gap between the vocabulary in our name tagger and the evaluation data. For instance, the pilot data contains new Organization names such as "Facebook", and "Twitter". To tackle this problem, we extracted a set of names with types of GPE, ORG, and PER from DBPedia[1], and a set of names from BBN IE results on the KBP 2014 source collection. To reduce the errors of the gazetteers, we removed names that contains non-English characters, and take the overlap between the two sets. The resulting gazetteers contain 32k PER names, 23k ORG names, and 20k GPE names.
- We encoded regular expression rules to extract posters as person name mentions.

2.2 Analyze Contextual Information of Mentions

We first analyze the original textual document and entity mentions to construct a document graph G_d. Depends on the analysis technique applied, different G_d can be constructed. In current implementation, we assume that if concept mentions are near to each other, connected by a short dependency path with special node types, a relation or an event, then these entity mentions are related to each other. An edge is added to two vertice if their corresponding entity mentions are related. We also leverage coreference information (Ng, 2010) to help us determine if a relation exists between two entity mentions. Figure 2 shows the document graph G_d for *Caldwell*.

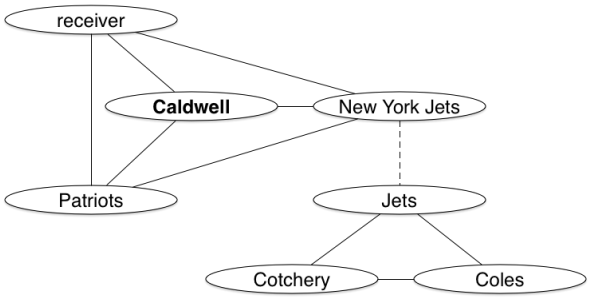

Figure 2: Document graph of Caldwell Example

By connecting the concept mentions to their related concept mentions, we construct graph G_d to represent the document and capture related information for each concept mention.

2.3 Entity Candidate Retrieval

By analyzing the triples in DBPedia describing the entities, we also construct a surface form dictionary $\langle f, \{e_1 \ , e_2 ... e_k\}\rangle$ where $\{e_1, e_2 ... e_k\}$ is the set of entities with surface form f. We analyzed the following main properties: labels and names (e.g. rdfs:label, foaf:firstName), redirect pages (e.g. wikiPageRedirects), and disambiguation pages (e.g. wikiPageDisambiguates), providing us with more than 100 properties to construct the surface form dictionary. During the candidate retrieval process, we retrieve all entities with surface forms that are similar to the mentions' surface form, and considered them as candidates for the mentions.

2.4 Non-Collective Entropy Rank

The candidate entities retrieved from the knowledge base are pre-ranked using an entropy-based non-collective approach (Zheng et al., 2014). The main idea of the algorithm is to assign the entities with higher popularity a higher score. While entities in Wikipedia are universally connected with the same type of link, entities in DBpedia are potentially connected with many kinds of links that may have

[1] http://dbpedia.org

semantically rich definitions. We can leverage this greater degree of specificity and assign different weights to edges described by different properties. For example, consider the triples (*_:Reche_Caldwell, _:formerTeam, _:New_England_Patriots*) and (*_:Reche_Caldwell, _:wikiPageLink, _:Tom_Brady*). Since "*New_England_Patiots*" and "*Tom_Brady*" are connected to "*Reche Caldwell*" by different relations, we consider their influence on the importance of "*Reche Caldwell*" to be different.

To capture such differences in influence, we compute the entropy of relations $H(p)$ (Shannon, 2001) as

$$H(p) = -\sum_{o_p \in O_p} \Psi(o_p) \log(\Psi(o_p)) \qquad (1)$$

where $p \in P$ is a property or relation that has a value $o_p \in O_p$ or links to an object $o_p \in O_p$ and $\Psi(o_p)$ is the probability of obtaining o given the property p. The entropy measure has been used in many ranking algorithms to capture the salience of information (Biesiada et al., 2005; Bruce and Tsotsos, 2005), therefore, in our task, we used it to capture the saliency of a property. In the previous example, p indicates "*formerTeam*" and "*wikiPageLink*" while *o* indicates "*New_England_Patriots*" and "*Tom_Brady*" respectively. Then $H(\text{“}formerTeam'')$ and $H(\text{“}wikiPageLink'')$ are the influence factors between "*Reche_Caldwell*" and "*New_England_Patriots*", and "*Reche_Caldwell*" and "*Tom_Brady*" respectively.

We then compute the salience score of candidate entities using the following non-collective EntropyRank:

$$ER(c) = \sum_{p^c \in P^c} H(p^c) \sum_{o_p^c \in O_p^c} \frac{ER(o_p^c)}{L(o_p^c)} \qquad (2)$$

where P^c is the set of properties describing a candidate entity c and $L(o_p^c)$ is number of entities linked to o_p^c. The EntropyRank for each entity starts at 1 and is recursively updated until convergence. This equation is similar to PageRank (Page et al., 1999), which gives higher ranks to the popular entities, but we also take the difference of influence of neighbor nodes into consideration.

As described previously, the candidate entities are retrieved from the surface form dictionary based on the above salience measure. Most often, the exact surface form match between an entity mention and a candidate entity cannot be found. However, our rank model allows partial surface form matches with a penalty. Currently we use Jaccard Similarity to compute partial match scores. For example, Jaccard Similarity will be computed for mention "nucleus" and entity "neural nucleus". In the equation below, $JS(m, e)$ is the Jaccard Similarity score between the surface form of entity mention m and the surface form of candidate entity c.

$$ER^*(m, c) = JS(m, c) \cdot ER(c) \qquad (3)$$

2.5 Collective Inference

In the non-collective inference approach, each entity mention is analyzed, retrieved, and ranked individually. Although this approach performs well in many cases, sometimes incorrect entity mention/entity links are formed due to the lack of context information. Therefore, we adopt a collective inference approach, which analyzes relations among multiple entity mentions and ranks the candidates simultaneously. For example, given the sentence that contains the entity mentions "*Reche_Caldwell*" and "*New_England_Patriots*", the collective approach will analyze the two mentions simultaneously to determine the best reference entities.

Using the connected G_d and candidate entities retrieved from the non-collective approach, we can compute the similarity between each entity mention m from G_d and a candidate entity c from G_k. Both m and c are connected to sets of neighbor nodes, which provide important contextual descriptions for both m and candidate entity c, respectively. We then use the following equation to compute the similarity score:

$$Sim^F(m, c) = \alpha \cdot ER^*(m, c) + \beta \cdot \sum_{p^c \in P^c} H(p^c) \sum_{n \in O_p^c \cap O^m} ER(n) \qquad (4)$$

Here, $O_p^c \cap O^m$ is the set of neighbors with equivalent surface form between the G_k subgraph for candidate c and G_d subgraph for mention m. The parameters α and β are used to adjust the effects of the candidate pre-ranking score and the context information score on the overall similarity score. Based on the

optimization results reported by Zheng et al. (Zheng et al., 2014), we empirically set $\alpha = 15$ and $\beta = 8$ for all experiments. The equation captures two important ranking intuitions: 1. the more popular a c is, the higher rank it will be, as captured by ER, 2. the more similar between the G_k subgraph for c and G_d subgraph for mention m, then higher rank will be given to c, which is captured by latter part of the equation.

To better describe the use of this system, we provide an illustrative example in Figure 3. For the example sentence provided, the document graph G_d has vertices V that correspond to entity mentions M. For this sentence-level collective inference approach, there exist edges between all vertices since these mentions co-occur in the sentence. Also the dot line represents the coreference relation between entity mentions. We then retrieve our knowledge graph G_k from our knowledge base. Focusing our attention on reference entity "*Caldwel*", a non-collective search returns candidates for "*Caldwell*". However, because "*Reche_Caldwell*" and "*Andre_Caldwell*" are connected to more vertices of G_k, it is intuitive that these candidates' rank increases with collective inference.

2.6 Collective validation

Collective inference provides a ranked candidate entity list for each entity mention that conforms to its local context information. However, the true meaning of the entity mention is constituted by the document context. To ensure the final selected candidate entity for each concept mention aligns with the document context, we validate and re-rank candidate entities of all mentions collectively.

First, we construct weighted graphs G_v from candidate graph G_c and G_d. If there is an edge between two concept mentions in G_d, then we check the KB to see if any candidate entities of these concept mentions are related. If relations are explicitly stated in the KB, an edge is assigned to the candidate entities. The computed information entropy of the relation is assigned as the weight for the edge. Also, if an explicit relation is not stated, but there is implicit evidence such that both candidate entities are related to certain entities, which suggest two entities are related, an edge is also added. In this case, the weight of the edge is proportional to the number of entities linked by both candidate entities. We then normalize both indirect co-occurrence weight and direct entropy weight to a value between 0 and 1. If there is no evidence suggesting any relation between c and any c', a stand alone node is added. Since we use G_d as a referent graph to construct G_cs, there is no redundant edge been added to G_c. We also process all candidate entities of all concept mentions, then the number of vertice and edge in all G_v is the same as G_d. For example, in Figure 4 we depict three possible G_v from G_c for "Caldwell" from many possible G_v. Then for each graph $G_v = \langle V, R \rangle$, we compute information volume (IV) using Equation 5.

$$IV(G_v) = \sum_{c_i, c_j \in V, p \in E} (Sim^F(m_i, c_i) + Sim^F(m_j, c_j)) * W(p) + \sum_{c_k \in V} Sim^F(m_k, c_k) \quad (5)$$

In this equation, c is the candidate entity which is a vertex in the graph G_c, p is the relation linking c_i and c_j. Sim^F is the computed similarity ranking score for c using Equation 4. We then select the candidate entity in the G_c that has the highest IV value. The first line of the equation computes the information of the graph base on the relations among the candidate entities. The second line of the equation computes the effect of similarity between mentions and candidate entities. The same intuitions we proposed are applied in this equation. If candidate entities are important, then the score of both parts of equation will increase. If the relation linking two candidate entities is strong or there are more relations in the graph G_d, then the score of the first part of the equation will increase. Using the equation, for G_vs in Figure 4, G_{v_A} is preferred among the three. Compare to G_{v_B}, G_{v_A} has stronger relation with *New England Patriots* with relation "formerTeam", where in G_{v_B}, relation between *Andre Caldwell* and *New England Patriot*.

2.7 NIL Clustering

Our Nil clustering approach is similar to our collective inference approach. For each unlinked entity mentions, we obtained the document graphs as previously described. We then compute the similarity

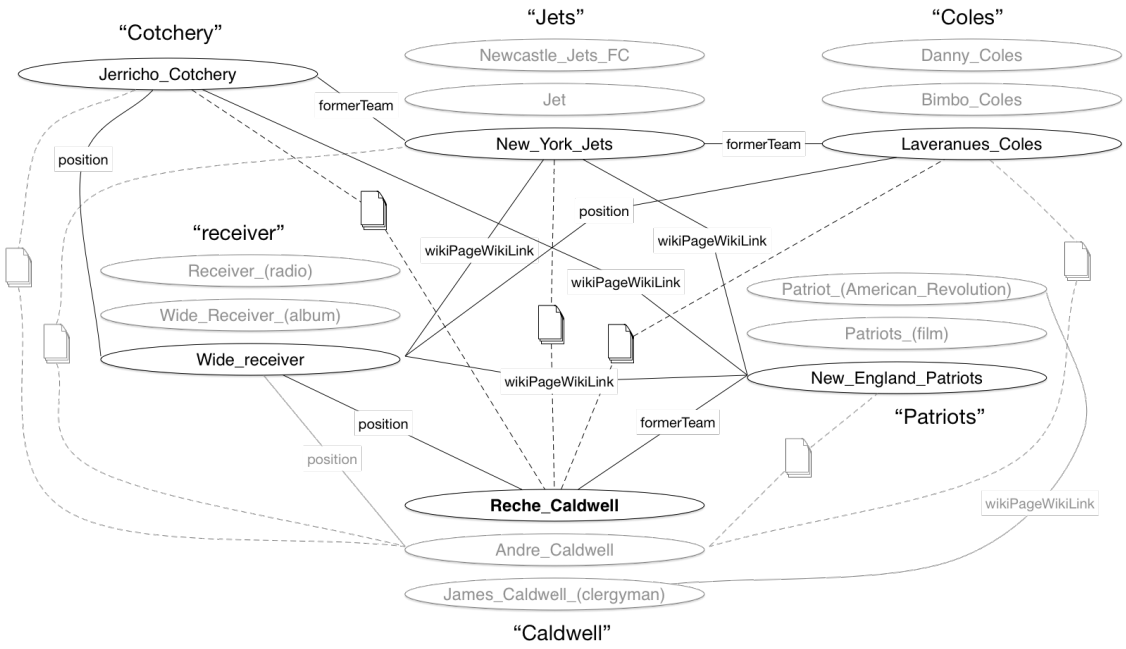

Figure 3: Candidate subgraph from G_k for the Caldwell example

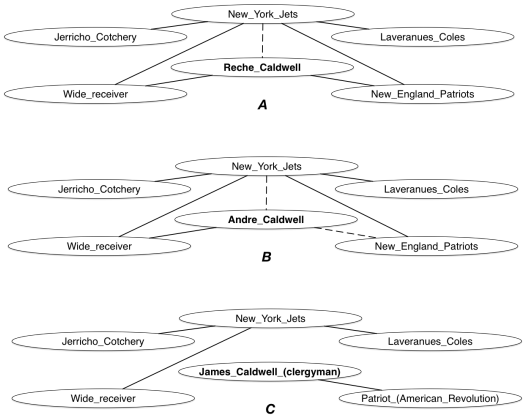

Figure 4: Example of possible G_v for the concept mention Caldwell

score between any two nil mentions' document graphs. The similarity score depends on two factors: 1. similarity between mention heads, for example, the string similarity between "Disney" and "Roy Disney". 2. contextual information similarity, for example, "Disney" may have contextual information such as "Company", "entertainment", "Florida", while "Roy Disney" has "Person", "1972", "father". For both string similarity and contextual information, we compute the Jaccard similarity and take the average of the two as our final similarity score. If the final similarity score is above a certain threshold (0.2 for our current implementation), we assign the mentions to the same cluster.

3 Regular Slot Filling

Our slot filling system consists three interdependent models: 1) Retrieve relevant documents to the target entities (Entity Search), 2) Detect the provenance of slot fillers in the relevant documents (Justification Identification), and 3) Extract the fillers from the justifications (Filler Extraction), similar to last year's system (Yu et al., 2013). We mined a lot of additional trigger phrases from NELL (**?**). The main change is Entity search, in which we improved relevant document acquisition through restrict fuzzy matching technique. More importantly, we added a temporality-based clustering model (TBCM) and a self-adapted active learning algorithm to the post-treatment of noisy search result detection and filtering.

3.1 Entity Search

Entity search is used to acquire the relevant documents of a specific entity mention from a large scale cross-media text data, such as the pre-assigned data

in TAC-KBP 2014, consisting of news stories, web documents and forum posts (nearly 2.1M documents in total). Relevant documents contain richer background information of the target entity and therefore can help discover the provenance of the relation between an entity mention and a slot filler, which is especially important for a slot filling system to extract eligible fillers and determine the text spans which justify the fillers (i.e., justification).

3.1.1 Basic Entity Search Engine

We built a local entity search engine by using the Apache Lucene of version 4.6, where some built-in query formulation and matching rules enable the exact search, namely purely fuzzy search or restricted fuzzy search over large-scale textual documents. To acquire relevant documents for the target entity mention, the search engine matches the formulated query with the existing documents, and once a document meets one of the rules, the document will be determined to be relevant to the entity mention. The entity search engine is consisted of eight components: query formulation, exact match, fuzzy match, restricted fuzzy match, query reformulation, query expansion, temporality-based document clustering and self-adaption active learning.

Query Formulation A query is basically formulated with the segmented words in the target entity mention without any pre-processing (e.g., stemming or stop word filtering), such as the query formulated with the words "mark" and "fisher" for the entity name "Mark Fisher". In this case, stemming and stop word filtering normally result in wrong senses of entity names and have a negative influence on the acquisition of relevant documents. For example, stemming process will mistakenly convert "fisher" to "fish" and lead to a different sense. If using the stemmed query "mark fish", the entity search engine will miss the documents particularly discussed about "Mark Fisher" but alternatively recall a lot of documents about "fish".

Exact Matching Under the exact matching rule, a relevant document necessarily contains the intact entity name and keeps the original word order of the name. For example, for the entity name "Greenberg Smoked Turkey", if a document contains the name string, it will be determined to be relevant, otherwise irrelevant. Sticking to the rule, the documents which only involve the string "Greenberg Turkey" (incomplete), "Smoked Turkey, Greenberg" (out-of-order), or "Smoked Turkey wing of Greenberg" (superfluous) will be determined as irrelevant, although they are actually correct retrieval results for the query "Greenberg Smoked Turkey".

Fuzzy Match The usage of exact match undoubtedly decreases the recall of the relevant documents. However, if we alternatively employ an unrestricted fuzzy match, we unavoidably approve the documents which purely involve componential loss copy of entity name during the relevant document determination. In this case, a large number of noise will be introduced into the search results, causing a low precision although the recall can be inflated. For example, those documents about "Roger Greenberg" (a movie actor), "Tobacco in Turkey", and "Smoked Bar" could be mistakenly determined as the relevant results of the query "Greenberg Smoked Turkey".

Restricted Fuzzy Match To make a trade-off between the effects of exact and fuzzy match, we suggest using the restricted fuzzy matching technology in the entity search. The matching rule allows the query to match a fixed-length word string (e.g., a sentence or n adjacent words) on the condition that the string contains all the query words, regardless of whether or not the query words occur discretely in the string or have a different sort order. For instance, according to the restricted fuzzy matching, the document which has a text span "The smoked wild Turkey from Greenberg" is also determined as a correct answer for the query "Greenberg Smoked Turkey". The underlying hypothesis of the matching rule is that the diversity of linguistic pragmatics approves the short-distance splitting and reordering of the originally coherent words in presenting the same entity (see the examples in Table 1). In the paper, we empirically set the distance as 15 (n=30).

3.2 Query Modification

In addition, we introduce query modification into entity search, including query reformulation and expansion. The query formulation uses logic expression (such as logic OR) to establish the principle of co-occurrence of query words in relevant documents. On the other hand, query expansion uses

Original Mention	Variance	Justification
Eloise Spooner (PER)	Roger **Spooner**, a Georgia farmer, and his wife **Eloise**	Roger **Spooner**, a Georgia farmer, and his wife **Eloise** tell their story of how ousted USDA official Shirley Sherrod helped them save their farm.
High Point Church (ORG)	Find **High Point**, North Carolina NC churches in our online **church**	Find **High Point**, North Carolina NC churches in our online **church** directory with address, phone number, website, denomination, map and more details.

Table 1: Short-distance splitting and reordering of the coherent words in presenting entities.

the keywords in the pre-assigned golden relevant document to expand the query, with the aim to take into account the background information of the entity during document relevance determination.

Query Reformulation An entity is often named in different ways and has multiple alternative names. Motivated by the finding, we enable the entity search engine to recall relevant documents which contain the alternate name. Some alternative name examples are presented in Table 2. We used two alternative name dictionaries mined from Wikipedia and DBPedia.

Obviously, it is necessary to add the identified alternate names to the query beforehand. Otherwise the entity search engine cannot detect the documents which contain the alternate names through any kind of match rule. In other words, the query should be the combination of the original and the alternate names. But it is difficult to formulate the query with diverse entity names. To solve the problem, we introduce the regular logic expression for query formulation, specially using the logic OR to manipulate the co-occurrence of query words, where either the original (ORI) entity name or the alternate (ALT) entity name is regarded as an independent unit, formulated with the pattern "ORI" OR "ALT", such as "Benjamin Chertoff" OR "Ben Chertoff". To confirm the joint effect of ORI and ALT in relevant document acquisition, we add logic disjunction relation among different entity names to the matching rule. Through the usage of the disjunction relation, a document which involves ORI, ALT or both can be determined to be relevant.

Query Expansion Additionally, we use the pre-assigned sole relevant document (golden document) to expand the query, by which to involve the contexts of the entity mention into the entity description. We select the top frequently occurred key words of the golden document to expand the query, e.g., to expand the query "Mark Fisher" with the key word "photographer" to generate the query "photographer Mark Fisher". The expanded query is helpful to disambiguate entity names and simultaneously filter the noise caused by the disordered words in the name. In Table 3 we can see that the expanded query "photographer Mark Fisher" can effectively shield the matching process from the disturbance of the wrong entities "**Shrewbury business owner** Mark Fisher", "Carrie **Fisher** and **Mark** Hamill".

But for the task of slot filling 2014, instead of directly using query expansion in the matching rule-based entity search, we jointly use it with text similarity calculation as the post-processing to optimize search results. After reformulating the query with logic OR expression, the entity search engine retrieves the pseudo-relevant documents through restricted fuzzy matching and logic disjunction validation, and in the stage of post-processing, for each of the search results, we calculate the textual similarity to the expanded query and select the most similar ones as the final results.

To select the keywords, for either the golden documents or the ones in the list of pseudo-relevant search results, we use TF-IDF to weight the words in documents and select 5 most weighted words as the keywords. In the text similarity-based relevance determination, an intuitive quantitative threshold is

[1] http://frustratingfraud.blogspot.com/2006/11/chertoff-tangent.html

[2] http://bankrupt.com/misc/TPISharePurchaseAgreement.pdf

[3] http://www.wcvb.com/politics/shrewsbury-business-owner-mark-fisher-running-for-governor/23370998

[4] http://screenrant.com/carrie-fisher-mark-hamill-getting-fit-star-wars-episode-7/

Original Mention	Variance	Justification
Benjamin Chertoff (PER)	**Ben** Chertoff	The article was principally by **Benjamin Chertoff**, popular Mechanics' 25-year of research editor... He says he called **Ben Chertoff** directly, and questioned the editor until...[1]
Thai Petrochemical Industry **Plc** (ORG)	Thai Petrochemical Industry **Public Company Limited**	Thai Petrochemical Industry **Public Company Limited**. The signing... Finance office, Thai Petrochemical Industry **Plc**. (TPI) by the...[2]

Table 2: Examples of a person's abbreviation and an organization's abbreviation.

Photographer Mark Fisher	**Business owner** Mark Fisher[3]	Carrie **Fisher** and **Mark** Hamill[4]
Mark Fisher American **Photogra-pher™**	**Business owner Mark Fisher** running for governor	Carrie **Fisher** and **Mark** Hamill are getting in shape for "Star Wars: Episode VII"

Table 3: The importance of contexts for entity disambiguation and noise filtering.

set as 0.1.

3.3 TBCM and Self-Adaption Active Learning

We propose a temporality-based clustering model (TBCM) to improve the query expansion-based entity search. The model is specially used to abstract the biography of an entity, with the ability of separately describing important segments of a life cycle. Through the model, the entity search can find the relevant documents happened around a historical moment and the relevance among the documents can be measured at the level of a fine-grained topic. To support the utilization of the model, we propose a TBCM based self-adaption learning algorithm. The algorithm firstly uses some precise search results as seeds to build the biography, such as the clusters of the pseudo-relevant documents retrieved by exact search and then iteratively searches the relevant documents and expands the biography from the highly-recalled search results, such as the ones in the search results returned by the fuzzy search.

Temporality-based Clustering Mode TBCM is used to avoid the bias in the process of query expansion. Normally, the relevant information of an entity should be consisted of various historical snapshots of the whole lifecycle (or named lifecycle segments), such as person's (PER) birth or organization's (ORG) foundation, marriage (PER) or consolidation (ORG), children (PER) or subsidiary (ORG), death (PER) or dissociation (ORG), etc. However, a few document (except the ones in Wikipedia) is able to summarize all the snapshots. Thus the expanded query which is generated by solely using the golden document most likely provides a partial expression of entity attributes, and accordingly the recalled relevant documents should concentrate on a small number of historical snapshots. Undoubtedly, such documents are helpless in exploring the fillers for all kinds of slot types of the entity. Therefore, it is ideal to obtain the seeds of historical snapshots and use them as leverage to find richer relevant information around various historical moments during the whole lifecycle.

The underlying hypothesis of TBCM is the seed-central cohesion of relevant documents along timeline, where each seed is a known relevant document released at specific time, recording a historical snapshot of an entity, and the most relevant documents to the seed should be released around the time and concentrate on the same topic in content. Abided by the hypothesis, TBCM previously detects the seeds and segments the timeline according to the released time of the seeds, where any boundary of a time segment is the intermediate time between two temporally nearest living seeds (see the vertical dotted lines in Figure 5). For the documents in each time segment, TBCM calculates the text similarity with the seed in the duration (see diagrams (a) and (b) in Figure 5). And then TBCM clusters the most similar documents around the seed (see diagram (b)

and (c) in Figure 5). As a result, TBCM can obtain the clusters of relevant documents to all previously known seeds that spread over the whole timeline, and if the seeds records different historical snapshots of an entity, the obtained relevant documents will not be limited to a historical moment of the entity. But here is a question that if the task of slot filling allows the usage of only one known relevant document to the entity, where should we find the seeds? We will answer this question in the self-adaption active learning algorithm.

a Temporal Segmentation (boundary is the intermediate time between temporally adjacent living seeds, e.g., the vertical dotted lines)

b Calculate text similarity between living seed and candidate in each time segment. The most similar candidate will be added to the clusters with the seed as kernel. Simultaneously the candidate is set to be the new living seed and used for the subsequent clustering.

c Iteratively cluster documents, deploy living seeds and slice time until none of new similar document is found for the living seeds.

d Each cluster consists of the initial seed and the newly found seeds. (The new seeds are none other than the most similar documents to the living seed in all the stages of the iterative clustering. See (b) and (c)).

Self-Adaption Active Learning We employe the query reformulation and the restricted fuzzy matching rule to obtain the relevant documents to a query mention. Either the restricted matching or the query expansion method is helpful to insure the correctness of the retrieved documents to some extent. Therefore we regard such documents as seeds. On the basis, we use the fuzzy match to recall as many candidate relevant documents as possible, where there are large-scale noises. Through the TBCM, we mine the relevant documents to the seeds from the candidates and use them to expand the seeds, by which we perform a round of self-adaption relevance learning. We perform the learning process iteratively until it meets a termination condition. The iteration ends at the time when the average similarity of the newly founded relevant documents to the seeds is lower than a presupposed threshold.

For the 50 entity queries (25 persons and 25 organizations) released in this years Slot Filling evaluation, the entity search system configured with the TBCM model and the self-adapted active learning achieves an F-score of 77%, along with a precision of 80% and a recall of 74%.

4 Event Argument Extraction

The core algorithm of our Event Argument Extraction system is based on our previous work (Li et al., 2013) and (Li et al., 2014). In (Li et al., 2013), we proposed a joint framework that extracts event triggers and argument links with a variety of global features. Let $x = \langle (x_1, x_2, ..., x_s), \mathcal{E} \rangle$ denote the sentence instance, where x_i represents the i-th token in the sentence and $\mathcal{E} = \{e_k\}_{k=1}^{m}$ is the set of argument candidates. Our framework extracts the best configuration $\hat{y}$ of event triggers and their argument links by using beam search such that:

$$\hat{y} = \arg\max_{y' \in \mathcal{Y}(x)} \quad \mathbf{w} \cdot \mathbf{f}(x, y')$$

where $\mathbf{f}(x, y')$ represents the feature vector for instance x along with configuration y'. The parameters are estimated by structured perceptron (Collins and Roark, 2004) algorithm with early-update. Within this framework, we can easily incorporate various global features. For instance, in the example in Figure 6, we can model the co-occurrence between *Die* trigger "died", and *Attack* trigger "fired". and further capture the phenomena that some arguments, such as *Place*, are shared by multiple triggers. (Li et al., 2014) extends this framework to a segment-based approach, in order to extract entity mentions, relations and events all together from scratch. In addition, this work proves that FrameNet is an informative resource to generalize event triggers, and gave a significant improvement in the experiments.

However, this work did not include temporal expressions, ACE values, and mention types. Therefore we incorporated the FrameNet based features into the system described in (Li et al., 2013), and use the IE system described in (Ji and Grishman, 2008) and the name tagging system described in (Li et al., 2012a) to provide event argument candidates and their coreference chains.

One difference between the EAE task and the traditional ACE event extraction is that EAE re-

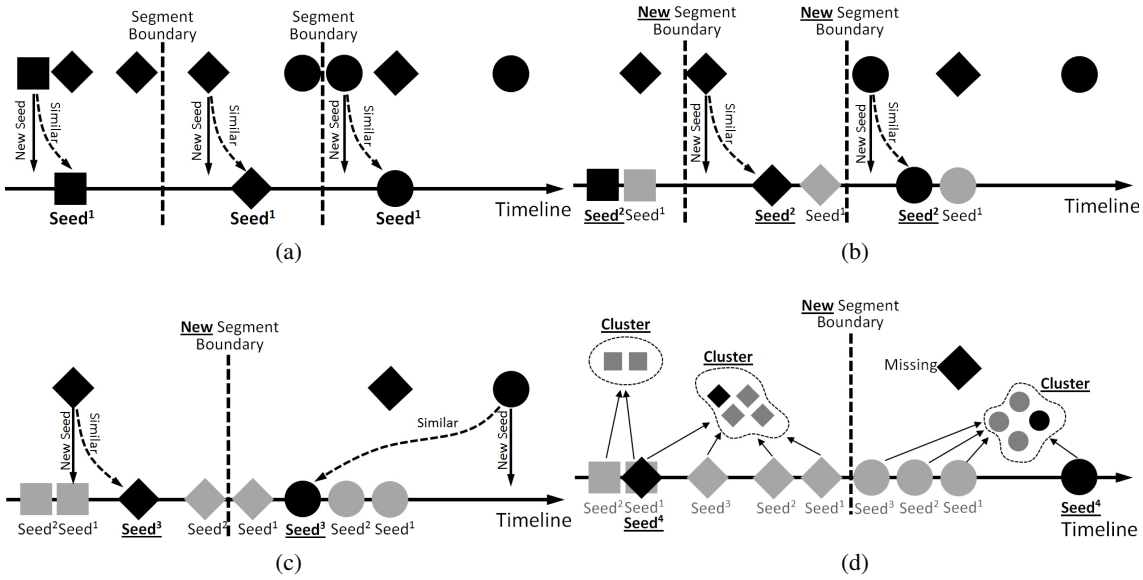

Figure 5: Basic steps of TBCM, including similarity calculation between seed and candidate document, temporal segmentation between living seeds, seed set expansion, and document clustering for historical snapshot finding (Note: the black solid geometries denote the living seeds in a stage of the clustering, while the gray ones are the initial and/or newly found seeds in previous stages)

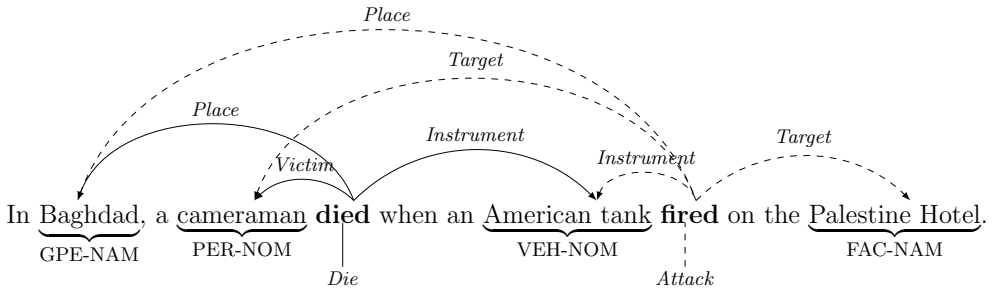

Figure 6: Example of joint event trigger and argument extraction.

Realis Label	ACE Properties
Actual	Tense = Past or Unspecified Polarity = Positive Genericity = Specific
Generic	Genericity = Generic
Other	All other cases

Table 4: Mapping table between ACE properties and Realis labels.

quires a system to produce a "*Realis*" label to each response. "*Realis*" has three possible labels: *Actual* is for events actually happened with the argument; *Generic* is for general events and arguments, and *Other* is for other cases.

To build a machine-learning based Realis classifier using available human annotations, we train four MaxEnt classifiers for *Modality*, *Polarity*, *Genericity* and *Tense* from ACE'05 corpus, respectively, apply the models to each event mention, and then map them to one of *Realis* labels based on the mapping table in Table 4.

The features for each classifier include trigger word, context words, pos tags, and property-specific word lists (such as "*may, can*", etc). These features are described in (Chen and Ji, 2009). We skip the detailed description here. The performance of those statistical classifiers are far from perfect. For example, (Chen and Ji, 2009) reported 0.63 F-measure of Modality, and 0.78 F-measure of Tense with system-output event mentions. To further improve the accuracy, we developed a set of rules based on lexical and syntactic information of each response. We utilized the word categories developed by (Li et al., 2012b) to capture indicative words. For example, *Consideration* category includes "*like*", "*consider*", and "*imagine*", and *Subjective* category includes "*assumed*" and"*supposed*". Table 5 summarizes the rules that we developed. At the decision time, we first apply MaxEnt classifiers to predict the Realis labels, then override the label by the rule-based prediction if any rule matches the response. In the pilot data, the rule-based component effectively reduced the number of responses with incorrect realis labels from 326 to 133.

Acknowledgments

Thanks to Hao Liu, Siyuan Ding, Kai Wang, Shanshan Zhu and Liang Yao at Soochow University, and the visiting students Keythe Gentry and Linda Li to RPI for their help on system and resource development. This work was supported by the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA), U.S. NSF CAREER Award under Grant IIS-0953149, U.S. DARPA Award No. FA8750-13-2-0041 in the Deep Exploration and Filtering of Text (DEFT) Program, U.S. Air Force Research Laboratory, IBM Faculty Award, Google Research Award, Disney Research Award, Bosch Research Award and RPI faculty start-up grant. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

Label	Rule
Other	the event mention is preceded by "*if*" the sentence includes "*guess*" or ends with "?" the sentence includes words in *Condition, Possibility and Negation* categories. the sentence includes words in *Subjective, and Consideration* categories.
Generic	the argument is preceded by "*any, most, every*" the argument is a plural noun without any modifiers pos(trigger word) = NNS (such as "*appointments*") pos(trigger word) = VBG, and it's not preceded by any of "*the, is, am, was, are, were, 's*" pos(trigger word) = VBZ or VBP pos(trigger word) = NN and the pos tag of the word before trigger is VBZ or VBP

Table 5: Rules for Realis labels. pos denotes part-of-speech tag.

References

J. Biesiada, W. Duch, A. Kachel, K. Maczka, and S. Palucha. 2005. Feature ranking methods based on information entropy with parzen windows. In *International Conference on Research in Electrotechnology and Applied Informatics*, volume 1, page 1.

N. Bruce and J. Tsotsos. 2005. Saliency based on information maximization. In *Advances in neural information processing systems*, pages 155–162.

Z. Chen and H. Ji. 2009. Language specific issue and feature exploration in chinese event extraction. In *Proc. Human Language Technologies: The 2009 Annual Conf. of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'09)*, Boulder, Colorado, June.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *ACL*, pages 111–118.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL*, pages 254–262.

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

Q. Li and H. Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proc. the 52nd Annual Meeting of the Association for Computational Linguistics (ACL2014)*.

Qi Li, Haibo Li, Heng Ji, Wen Wang, Jing Zheng, and Fei Huang. 2012a. Joint bilingual name tagging for parallel corpora. In *21st ACM International Conference on Information and Knowledge Management*.

Xiang Li, Heng Ji, Faisal Farooq, Hao Li, Wen-Pin Lin, and Shipeng Yu. 2012b. Rich annotation guided learning. In *International Journal On Advances in Intelligent Systems*.

Q. Li, H. Ji, and L. Huang. 2013. Joint event extraction via structured prediction with global features. In *ACL*, pages 73–82.

Q. Li, H. Ji, Y. Hong, and S. Li. 2014. Constructing information networks using one single model. In *Proc. the 2014 Conference on Empirical Methods on Natural Language Processing (EMNLP2014)*.

Vincent Ng. 2010. Supervised noun phrase coreference research: the first fifteen years. In *Proc. 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1396–1411.

L. Page, S. Brin, R. Motwani, and T. Winograd. 1999. The pagerank citation ranking: Bringing order to the web.

C. E. Shannon. 2001. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55.

M. Surdeanu and H. Ji. 2014. Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In *Proc. Text Analysis Conference (TAC2014)*.

D. Yu, H. Li, T. Cassidy, Q. Li, Z. Chen, and H. Ji. 2013. Rpi-blender tac-kbp2013 knowledge base population system description. In *Proc. Text Analysis Conf. (TAC'13)*.

D. Yu, H. Huang, T. Cassidy, H. Ji, C. Wang, S. Zhi, J. Han, C. Voss, and M. Magdon-Ismail. 2014. The wisdom of minority: Unsupervised slot filling validation based on multi-dimensional truth-finding. In *Proc. The 25th International Conference on Computational Linguistics (COLING2014)*.

J. Zheng, H. Wang, H. Ji, and P. Fox. 2014. Linkipedia: Entropy based collective entity linking with dbpedia. In *Technical Report, Rensselaer Polytechnic Institute*.