Intelius-NYU TAC-KBP2012 Cold Start System

Ang Sun, Xin Wang, Sen Xu, Yigit Kiran,
Poornima Shakthi, Andrew Borthwick
Intelius Inc.
500 108th Ave NE Suite 2200 Bellevue, WA

asun@intelius.com

Ralph Grishman

Computer Science Department
New York University

grishman@cs.nyu.edu

Abstract

This paper describes the Intelius-NYU 2012 system for the KBP Cold Start task. The Cold Start task can be decomposed into two subtasks: slot filling and entity linking. For slot filling, we focus on the adaptation of the NYU 2011 regular slot filling system to the Cold Start task. For entity linking, we apply Intelius's commercial conflation engine to link person and organization entities with minimal tuning for the Cold Start task. We also developed a voting-based entity linking system for geo-political entities.

1 Introduction

The Cold Start task aims to create a knowledge base from a large text corpus. To build such a knowledge base, a Cold Start system needs to have a slot filling component that could extract entities and their attributes from the corpus, which can be further used as evidences for linking these entities to the right nodes in the knowledge base.

In the next section, we describe the adaptation of the NYU 2011 regular slot filing system to the Cold Start task. Section 3 presents the details of the Intelius's conflation engine that is used to link person and organization entities. Section 4 describes our entity linking system for geo-political entities (GPEs). We present experimental results and error analyses in Section 5 and conclude in Section 6.

2 Cold Start Slot Filling System

In this section, we first briefly describe the NYU 2011 regular slot filling system. A regular slot filling system focuses on the extraction of attributes, or slot fills, for a given query entity from a large text corpus, while a Cold Start slot filling system needs to do such extraction for every entity occurring in a single document. To coordinate with such task requirements, we use the co-reference information of the entities in a single document to facilitate the extraction. An entity linking system usually needs more than just the entities and their attributes as evidences to make the linking decisions, we will describe the contextual information that are provided by our slot filling system.

2.1 The NYU 2011 Regular Slot Filling System

Figure 1 shows the architecture of the NYU system that was used for both 2011 and 2012. Like most regular slot filling systems, the NYU system has 3 basic components: document retrieval, answer extraction, and merging. Document retrieval returns a list of relevant documents for a given query entity. Then two extractors are applied to these documents to find the attributes of the entity: one extractor uses a set of classifiers trained with distant supervision and the other uses a set of patterns. At the end of the pipeline, the merging component validates answers and removes duplicates. For more details of the system, please refer to the NYU 2011 and 2012 system papers.

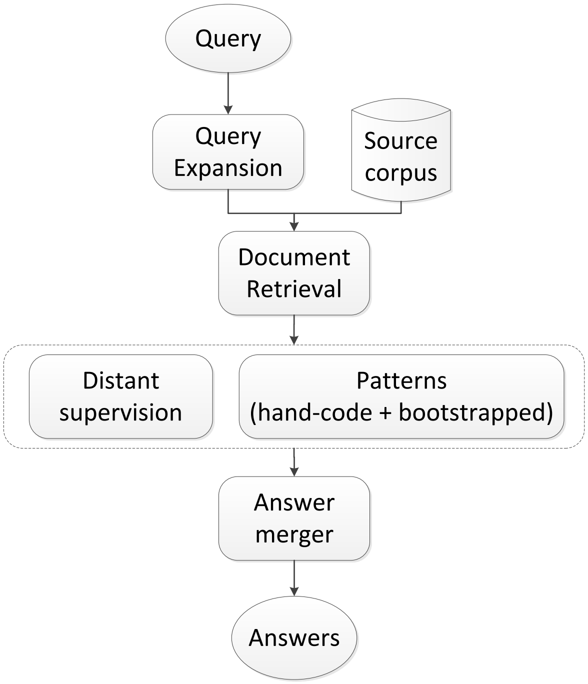
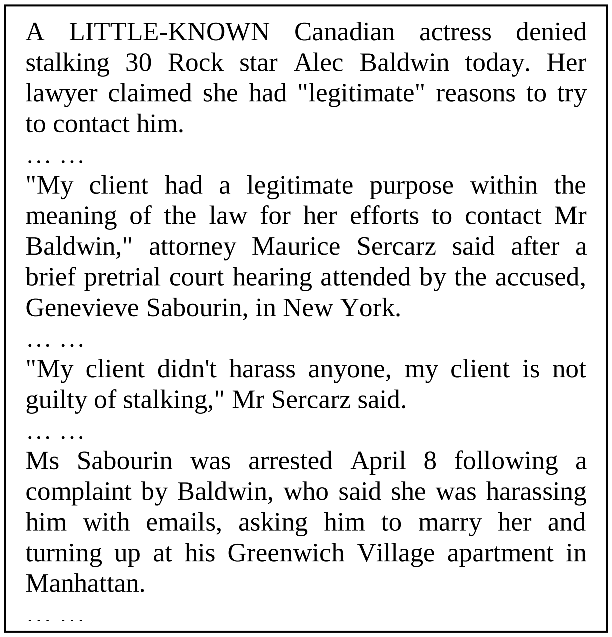


Figure 1: The NYU 2011 Slot Filing System

2.2 Within Document Slot Filling for Cold Start

Within document co-reference: In Cold Start, we need to do extraction for every single document in the given evaluation corpus. One important subtask here is to find the entities occurring in a single document. We use the NYU Jet[1] system to do within document co-reference to facilitate this subtask (Grishman et al., 2005). Formally, given a document *D*, the co-reference system outputs a list of entities <E_1, … E_n>. For example, given the sample document as shown in Figure 2, the co-reference system outputs 3 person entities <*Alec Baldwin*, *Maurice Sercarz*, *Genevieve Sabourin*> and organization and GPEs <*Canadian*, *New York*, *Greenwich Village*, *Manhattan*> as well.

Each entity is associated with a list of entity mentions of three types: name (Alec Baldwin), nominal (star), and pronoun (him). For named mentions, we pick the longest one as the *canonical mention*. For example, there are two named mentions of the entity *Maurice Sercarz*, "*Maurice Sercarz*" and "*Sercarz*". We treat "*Maurice Sercarz*" as the *canonical mention*. Nominal and pronoun mentions are used for extracting the attributes but are omitted from the Cold Start system output as defined by the task guideline (We refer the reader to the NYU system papers for the details of how these co-reference information were used for slot filling).



A LITTLE-KNOWN Canadian actress denied stalking 30 Rock star Alec Baldwin today. Her lawyer claimed she had "legitimate" reasons to try to contact him.

… …

"My client had a legitimate purpose within the meaning of the law for her efforts to contact Mr Baldwin," attorney Maurice Sercarz said after a brief pretrial court hearing attended by the accused, Genevieve Sabourin, in New York.

… …

"My client didn't harass anyone, my client is not guilty of stalking," Mr Sercarz said.

… …

Ms Sabourin was arrested April 8 following a complaint by Baldwin, who said she was harassing him with emails, asking him to marry her and turning up at his Greenwich Village apartment in Manhattan.

… …

Figure 2: A Sample Document

Given the *canonical mention* of an entity and the document it appears, reusing the NYU 2011 slot filling pipeline for the purpose of extracting entities and their attributes becomes straightforward. Note that the step of document retrieval is removed from the pipeline as the document is already given. The sample extraction for the entity *Maurice Sercarz* is shown in Rows 1 and 2 of Table 1.

Slot Filling for GPEs: The NYU 2011 system only extract attributes for person and organization entities. We did not develop a separate extractor for GPEs for this year's Cold Start. We instead infer their attributes from the extractions of person and organization entities. For example, we would infer a slot fill for the slot type *gpe:births_in_city* from a slot fill for the slot type *per:city_of_birth*.

Contextual Information Extraction: Besides names and attributes of the entities, the contextual information could be beneficial to the entity

[1] http://cs.nyu.edu/grishman/jet/license.html

linking system as well. Specifically, we output all the names extracted in the document by a named entity extraction model (Sun and Grishman, 2011) and the blurb of the entity. The blub extraction walks through the co-reference chain of the entity in the document and outputs every sentence that contains a mention of that entity (Bagga and Baldwin, 1998).

Entity	*Canonical Mention*: Maurice Sercarz *Mention*: Sercarz
Attributes	*per:title*: attorney
List of Names	*Person*: <Alec Baldwin, Baldwin, Maurice Sercarz, … > *Organization*: <> *Location*: <Canadian, New York, Greenwich Village, Manhattan, …>
Blurb	"My client had a legitimate purpose within the meaning of the law for her efforts to contact Mr Baldwin," attorney Maurice Sercarz said after a brief pretrial court hearing attended by the accused, Genevieve Sabourin, in New York. "My client didn't harass anyone, my client is not guilty of stalking," Mr Sercarz said. "

Table 1: Slot Filling Output for the Entity *Maurice Sercarz* in the Sample Document

3 Entity Linking for Person and Organization

In this section, we briefly describe Intelius' conflation pipeline and how we adapt it to the Cold Start Task.

3.1 Intelius Entity Linking Pipeline

At Intelius, we collected more than six billion people records from publicly available sources. To link all these records to the correct person entity, we developed a Map-Reduce based entity linking pipeline that ran on Hadoop based clusters (private or cloud based). Figure 3 gives a simplified overview of our conflation pipeline. It contains four stages: Blocking, Link Scoring, Clustering and Coalesce.

The Blocking stage uses different combination keys and hashing functions to group records that are more likely to be about the same person into the same block (McNeill et al., 2012).

The Link Scoring Stage uses a machine learning based model to score each possible link inside the block (Chen et al., 2011).

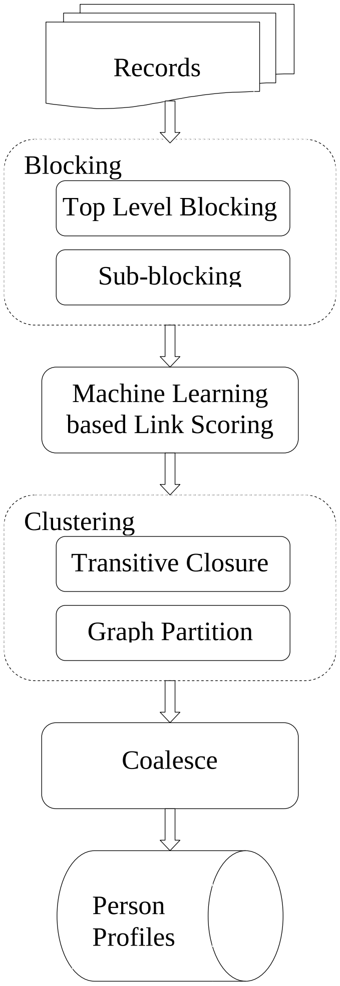


Figure 3: The Intelius Conflation Pipeline

The Clustering Stage builds bigger subgraphs that can be loaded into memory by linking smaller blocks together transitively. Then, the subgraphs are partitioned into clusters according to the scores of their links.

The Coalesce Stage reshuffles the records stored on the Hadoop File System based on clustering results, and merges records into single person profiles by reconciling information from different sources. In this Stage, the pipeline collects the values of a slot from the records in a cluster, converts them to their canonical forms, and removes the duplicates.

Each stage is highly parallelized and is minutely optimized to process billions of records in a linearly scalable way. On an 87-node commodity hardware based cluster, it takes the pipeline less than a week to link three billion records to about 600 million person entities.

3.2 Adaption For Cold Start Person and Organization Entity Linking

Before the Cold Start task, the Intelius conflation pipeline was optimized to have high precision and recall on public person records. To run it on local news and web blogs, we had to adapt the Blocking and the Link Scoring stages of the pipeline to these domains.

Blocking

The Blocking Stage has two phases, the Top-Level Blocking phase and the Sub-Blocking phase. The Top-Level Blocking phase groups records into overlapping 'blocks'—sets of records that are likely to be the same person based on values of the blocking keys. Blocking keys can be attributes of an entity such as name, location, email, employer, and relative, and the combinations of these attributes. Or they are mapping or hashing functions that turn a set of attributes or the entire record into a set of values.

The sizes of these blocks can be huge if the Top-Level blocking keys do not have enough granularities. For example, if using the pair of attributes <*First Name, Last Name*> as a blocking key, the block for the key value <*John, Smith*> can have millions of records.

If the blocks are too big, the Sub-Blocking phase of the pipeline will try more granular blocking keys (Sub-Blocking keys) to divide the blocks even further. The Sub-Blocking process will continue until the blocks are small enough.

In Table 2, at the beginning of each block, there is a list of blocking keys separated by '|'. The first one is the Top Level Blocking key, and the rest are the Sub-Blocking keys. For example, in the first block, the blocking keys are *REFERENCE LIBRARIAN* | *PETROVSKY*, *REFERENCE LIBRARIAN* is the Top Level Blocking Key, *PETROVSKY* is the sub-blocking key.

News and Web-blogs often refer to person entities by nicknames or informal forms of their first names (*Bill Clinton* for *William Clinton*, *Jim Carter* for *James Carter*). To link these entities together, first, the Blocking Stage needs to put them into the same block. We build equivalent classes of nicknames, and use them in customized blocking keys that map records with the names *Jim Carter*, *Jimmy Carter* and *James Carter* into the same group *James Carter.*

The same is true for titles of people and names of organization. *Software Engineer* and *Computer Programmer* refer to more or less the same type of job. *IBM* is the acronym for *International Business Machines*. To solve this problem, we build equivalent classes of job titles and organization names and use them as blocking keys.

More interestingly, people's job title can change over time. For example, *Senator Hilary Clinton* becomes *Secretary Hilary Clinton* after the 2008 Presidential Election. To capture these cases, we build a gazetteer for equivalent classes of job titles by following the nickname equivalent class generation algorithm as described in (Carvalho et al., 2012). Typical clusters of job titles look like:

STAFF PHARMACIST,
PHARMACY SUPERVISOR
PHARMACY PRACTICE RESIDENT
PHARMACY STUDENT
CLINICAL PHARMACY COORDINATOR
RELIEF PHARMACIST
CLINICAL STAFF PHARMACIST
CLINICAL PHARMACIST ASSISTANT
PHARMACY MANAGER
DPHARM STUDENT
SR PHARMACY TECHNICIAN
PHARMACY DISTRICT MANAGER
PHARMACY INTERN
LEAD PHARMACY TECHNICIAN
PGY PHARMACY PRACTICE RESIDENT
PHARMACY EXTERN STUDENT
… …

Unlike in the Coalesce Stage, we don't need to find the exactly correct clusters of nicknames, titles and organization name clusters. As long as the related records can be put into the same block, and the size of the block does not get bloated to a point that it can't be processed by the pipeline, we can trade precision for recall.

Attributes of entities that are being extracted are usually sparse and the conflation of our system depends heavily on the contextual information extracted from the same article (Person, Organization, GPE mentions and the blurb). To address this issue, we developed TF-IDF based hash functions on these contextual information to use as Sub-Blocking keys.

Table 2 shows an example set of blocks with their blocking keys for person entity linking from one of our runs on the Cold Start Task data set:

REFERENCE LIBRARIAN|PETROFSKY E11888 E11850 E11892 E11855
PROFESSOR OF CHEMISTRY|HOCHSTRASSER E316915 E314289
PROFESSOR ENGLISH|ROZIN E222807 E223538 E31125 E37922
PROFESSOR ENGLISH|POWELL E17891 E17870 E37343
CHRIS |PASTORE|LPS WEBSITE |DIRECTOR| 2012 COLLEGE OF LIBERAL|MAIN PENN|LPS E74211 E74066 E74198 E74375 E74175 E74312 E74276 E74463 E74353 E74112 E74299 E74451 E74507 E74519 E74439 E74531 E74023 E74397 E74099
YVETTE|BORDEAUX|2012 COLLEGE OF LIBERAL|DIRECTOR|LPS WEBSITE|MAIN PENN|LIBERAL ARTS PROGRAM E821375 E821471 E821071 E821035 E821459 E821387 E821059 E821047

Table 2: Example blocks. Words with the initial E are indices of person entities. Words in bold are blocking keys which can be the Name, Location, Job Title or Tf-idf keywords collected from Person, Organization and GPE mentions, and blurbs in the article.

Link Scoring

The Link Scoring model in the Intelius's conflation pipeline is an Alternating decision tree based supervised model that operates over a pair of records. A pair of records is described by a vector of various features:

- Name frequency, location, population features, and features that combine all three together
- Features comparing and linking other attributes/relations of people

For example, *RegionalNBP* is one of the features that combine name frequency, location, population information from two records, and decides whether the two records are about the same entity. First, it checks the location attributes, and see how many regions the two records have in common. By region, we mean a country, a state, a city, a metropolitan area, a county, or even a neighborhood inside a city. For each common region, it checks the frequency of the name in that region, and the population of that region, and then it computes an approximate likelihood of another person has exactly the same demographic characteristics. Finally it compares and combines these likelihoods into a final likelihood number to decide whether the two are the same person.

To adapt our Link Scoring model to the local news and web blog domain, we added two new sets of features:

- Features comparing and linking among KBP specific attributes
- Tf-idf and N-gram features for contextual information from the articles

In total, we used 116 features, and about 50,000 training examples collected from social network, news and other similar data sources.

For linking organization entities, we used the following groups of features:

- Location based
- Features comparing and linking KBP specific attributes
- Ngram and Tf-idf based features

In total, we used 60 features and about 4,000 training examples collected from profiles extracted from previous years' KBP corpus.

4 Entity Linking for Geo-Political Entity

The challenge in entity linking for GPE is that GPE names can be quite ambiguous. In Figure 2, *New York* could be referring to the State or the City. Famous ambiguous GPEs include *China* (Country or Town in Maine, US), *Georgia* (Country or State in the US), *Springfield* (common City name appears in more than 10 US States), *Berlin* (Capital of Germany, also a State in Germany, also a common City name in the US). Using the Geonames database[2], we found over 5,000 ambiguous geonames. We developed a voting-based system using contextual GPEs to achieve disambiguation.

4.1 GPE Disambiguation

Disambiguating toponyms has been an active research topic in the Geographic Information Science community (Buscaldi and Rosso, 2008; Garbin and Mani, 2005; Zhang et al., 2012; Overell and Rüger, 2008). Buscaldi and Rosso

[2] http://www.geonames.org/

(2008) used contextual toponyms and external geographical databases to calculate the cumulative distance among all toponyms appears in one document. For ambiguous toponyms, they chose the candidate that yielded minimal cumulative distance with the contextual toponyms. The assumption is that toponyms that are mentioned together should be relatively near spatially. We took a similar approach: instead of calculating spatial distance, we used a GPE hierarchical relationship to achieve disambiguation.

First, we define a hierarchy of GPE types: *Country* -> *Province (State)* -> *City*. This hierarchy is used to disambiguate GPEs, since ambiguous GPE names either a) are of different types (e.g., *China* as a Country is different in type from the City *China* in *Maine, USA*; or b) differ in terms of their higher-level types. For example, *Austin, TX* and *Austin, MN* share the same name at the City level, but they can be separated at the level of Province/State. In this hierarchy, co-occurrence of siblings (*Austin* mentioned together with *Houston, both are city names in Texas*) or directly parental relationship (*Austin* mentioned together with *Texas*) will be rewarded, making *Austin, TX* the more probable candidate.

Second, a gazetteer is built from the Geonames database in order to assign the above types to extracted GPEs. The gazetteer is designed as a unique toponym index that includes toponym, upper hierarchy toponym and population (see Table 3).

The disambiguation algorithm works as follows:

```
Given GPEs extracted, find all matches from the Gazetteer.
if a GPE is ambiguous (have more than one candidate)
    if there are no contextual GPEs,
        return the most populated candidate;
    else if there are contextual GPEs,
        go through the voting scheme to find the most likely candidate.
Return a list of unique GPEs, together with additional geographic information (population, Country and State information for City, Country information for State) that can be used to assist linking person and organization entities.
```

To illustrate this workflow, we will walk through a disambiguation example of the following GPEs: <*Bellevue, Washington, Seattle, U.S.*>. Seattle and U.S. are unique toponyms in our gazetteer, so the lookup will return:

Seattle: City_InState_Washington_InCountry_US
U.S.: Country

Washington and Bellevue are two ambiguous toponyms (for illustration purpose, only two candidates are described here)

Washington: State_InCountry_US; City_InCountry_UK; …
Bellevue:
City_InState_Washington_InCountry_US;
City_InState_Nebraska_InCountry_US; …

For each ambiguous toponym, each of its candidates will go through the contextual GPEs for votes. For *Washington*, the contextual GPEs (*Seattle, U.S., Bellevue*) will vote for it to be as a US State or a city in UK: because *Seattle, Bellevue* can be offsprings of *Washington* as a US State (upvotes) and *U.S.* is the direct parent of *Washington* as a US State (another upvote). *Washington* should be recognized as the type State. For *Bellevue*, Contextual GPE *Seattle* will vote for *Bellevue* to be a city_InState_Washington as they are siblings (upvote); *Washington* will vote the same due to parent-ship (another upvote); *U.S.*, as grandparent for both candidate (*City_InState_Washington* and *City_InState_Nebraska*) will vote for both. In the end, Bellevue as a *City_InState_Washington* gets a higher vote.

Key	Value
China	Country_POP_1,330,044,000 City_InState_Maine_InCountry_US
Seattle	City_InState_Washington_InCountry_US
Georgia	Country_POP_4,630,000 State_POP_8,975,842_InCountry_US
…	…

Table 3. Gazetteer Sample

4.2 Beyond Disambiguation

The gazetteer provides GPE types as well as population and parental information (the state that a city belongs to, the country that a state belongs to). This additional information could serve as features for conflation for person and organization entities. When an entity is correctly associated with a GPE, the inferred association between the entity and the GPE's parents is also a valuable feature to consider for further processing. For example, "Obama was born in Honolulu" may yield: Person: <Obama>, GPE: <Honolulu>, Relationship: <born_in>. After the disambiguation step, *Honolulu* is referring to *City:Honolulu, HI, US* which could be used to infer the person's born_in city, state, and country. This is one of our future works.

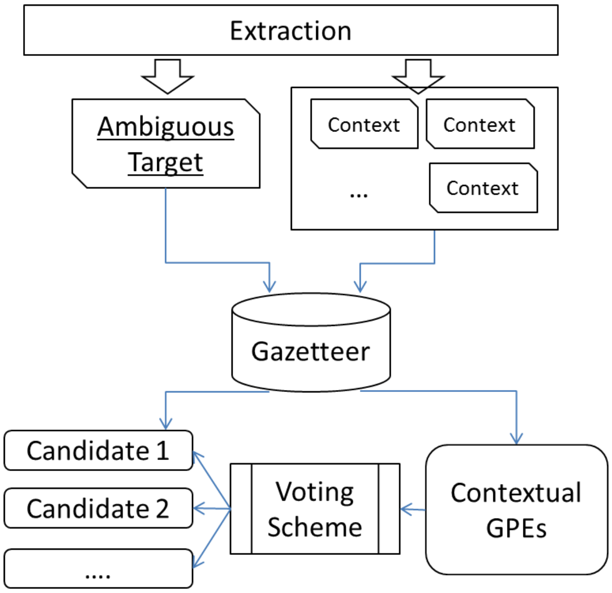


Figure 4: Workflow for GPE disambiguation. The voting scheme checks how each contextual GPE relates to a candidate. If there are siblings (city names within the same state, state names within the country) or direct parent relationship (contextual GPE is a state that candidate city belongs to), the candidate will get a higher vote.

5 Experiments

Even though scalability is uber important to build a knowledge base from web-scale text corpuses, in our experiments, we found that the system is an overkill for this year's Cold Start Task. In Run 2, we did conflate the entities extracted from the Cold Start Task Text Corpus against our commercially available people profiles (600 millions), but the results were not as good as we have thought. In addition, to scale the entity linking system to billions of records, we had to make architectural and algorithmic compromises that hurt its accuracy and performance when executed on a much smaller text corpus. Moreover, the Cold Start evaluation domain is about university sites, many of which are not always news articles.

But after the competition, we successfully incorporated the components we developed for the Cold Start task into our commercial Local News Information Extraction and Conflation Pipelines. It successfully processed 180 million local and national news articles in one day, extracted 380 million entities, about half of which were linked to our commercially available profiles.

From this production run, we randomly sampled 3,587 linked pairs (a pair contains an Intelius person profile and a person profile from news) and put them up at Amazon Mechanical Turk to have them labeled by Intelius Super Turkers (Turkers who have worked with us before and have an average correct rate that is above 75%). After that our internal data raters relabeled the pairs on which the Super Turkers disagree. This evaluation shows that 90% of these linked pairs were correct.

6 Conclusion

We have described the Intelius-NYU 2012 system for the KBP Cold Start task. We focused on two system adaptation tasks for this year's Cold Start: adaptation of the NYU 2011 regular slot filling system to the Cold Start slot filling subtask and adaptation of the Intelius's commercial conflation engine to the Cold Start entity linking subtask for person and organization entities. We also developed a voting-based entity linking system for geo-political entities.

Acknowledgments

References

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In Proceedings of COLING-ACL, 79-85.

Davide Buscaldi & Paulo Rosso. 2008. Map-based vs. knowledge-based toponym disambiguation. 2008. In *Proceeding of the 2nd international workshop on Geographic information retrieval*, GIR '08 (pp. 19–22). New York, NY, USA: ACM.

Vitor R Carvalho, Yigit Kiran, Andrew Borthwick. 2012. The Intelius Nickname Collection: Quantitative Analyses from Billions of Public Records. In *Proceedings of NAACL-HLT 2012.*

Sheng Chen, Andrew Borthwick, and Vitor R Carvalho. The case for cost-sensitive and easy-to-interpret models in industrial record Linkage. 2012. In *Proceedings of the 9th International Workshop on Quality in Databases, 2012.*

Eric Garbin and Inderjeet Mani. Disambiguating toponyms in news. 2005. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 363–370, Stroudsburg, PA, USA, 2005.

Ralph Grishman and Bonan Min. 2010. New York University KBP 2010 Slot Filling System. In *Proceedings of Text Analysis Conference 2010.*

Ralph Grishman, David Westbrook and Adam Meyers. 2005. NYU's English ACE 2005 System Description. ACE 2005 Evaluation Workshop.

Bill McNeill, Hakan Kardes, Andrew Borthwick. 2012. Dynamic Record Blocking: Efficient Linking of Massive Databases in MapReduce. In *Proceedings of the 10th International Workshop on Quality in Databases, 2012.*

Bonan Min, Xiang Li, Ralph Grishman and Ang Sun. 2012. New York University 2012 System for KBP Slot Filling. In *Proceedings of Text Analysis Conference 2012.*

Mike Mintz, Steven Bills, Rion Snow and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP 2009.*

Simon Overell and Stefan Rüger. Using co-occurrence models for placename disambiguation. *International Journal of Geographical Information Science*, 22(3):265–287, 2008.

Sebastian Riedel, Limin Yao and Andrew McCallum. 2010. Modeling Relations and Their Mentions without Labeled Text. In *Proceedings of ECML/PKDD 2010*.

Ang Sun, Ralph Grishman, Wei Xu and Bonan Min. 2011. New York University 2011 System for KBP Slot Filling. In *Proceedings of Text Analysis Conference 2011*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, Christopher D. Manning. 2012. *Multi-instance Multi-label Learning for Relation Extraction.* Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning (EMNLP-CoNLL), 2012.

Mihai Surdeanu, Sonal Gupta, John Bauer, David McClosky, Angel X. Chang, Valentin I. Spitkovsky, Christopher D. Manning. 2011. *Stanford's Distantly-Supervised Slot-Filling System.* Proceedings of the TAC-KBP 2011 Workshop, 2011.

Xiao Zhang, Baojun Qiu, Sen Xu, Prasenjit Mitra, Alexander Klippel, and Alan Maceachren. Disambiguating and geocoding road names in text route descriptions using exact-all-hop shortest path algorithm. In *ECAI 2012*, 2012.