The NYU Cold Start System for TAC 2015

Yifan He **Ralph Grishman**
Computer Science Department
New York University
{yhe, grishman}@cs.nyu.edu

Abstract

The KBP Cold Start task builds a knowledge base from scratch using a given document collection and a predefined schema for the entities and relations. We describe the NYU submission to the TAC 2015 Cold Start track (KB variant). We report the overall architecture, new modules we introduce this year, experimental results, and experiments we conduct on building KBP relation extractors using an active learning tool developed for NLP novices.

1 Introduction

Knowledge base construction usually involves a large amount of human labor. The KB's are either curated by human annotators or acquired automatically from the text, which in turn requires sustained effort from experts in computer science and linguistics to annotate large amount of data and to develop sophisticated algorithms.

During this year's participation of TAC, we try to experiment with methods that can potentially allow domain experts - who are not necessarily trained NLP researchers - to bootstrap a KB extraction system rapidly. To this end, we start from NYU's KBP system used in previous years which relies heavily on a matcher on dependency paths between entities (Section 2), enhance this matcher to allow matching between dependency paths that are similar but not identical (Section 3), and finally experiment with replacing the dependency path rules in our KBP systems with rules that are extracted using ICE[1], an information extraction customizer intended to be used by NLP novices (Section 4).

2 System Architecture

Our submission is based on NYU's submission to the TAC Cold Start track in 2014 (Nguyen et al., 2014). We briefly review the pipeline here:

- **NLP processing.** We first run input documents through the Jet NLP pipeline[2]. We record noun phrases, named entities, dependency parse trees, and coreference chains.
- **Processing high frequency slots.** We use a rule-based "core tagger" to handle frequent noun-phrase internal relations, including titles and relatives. The logic of this tagger is hardcoded into the system.
- **Rule-based relation extraction.** For most slots, we rely on a set of lexical and dependency path rules to find the correct response. Given a name and its potential slot fillers, we check if the lexical or dependency path between the name-filler pair matches a rule in our rule set. If there is an exact match, we record the slot fill.
- **Distantly supervised tagger.** For name-fill pairs that do not match an extraction rule, we run them against a distantly supervised MaxEnt relation classifier. The classifier was trained on the TAC 2010 document collection, in which relation instances were annotated by aligning

[1] http://nlp.cs.nyu.edu/ice/

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

the text to Freebase relation tuples (Sun et al., 2011).

- **Combination.** After every document is processed, we perform cross-document coreference to combine the output and construct a KB. Cross-document coreference is based on string matching.

During the development of this year's system, we introduce miscellaneous improvements to various components, including coreference resolution and forum post detection. We will incorporate them into future Jet releases.

We make most of the changes this year with regard to the dependency rules in the **rule-based relation extraction** step: we further edit the set of relation extraction rules manually, as we tuned our system against TAC 2014 evaluation data, allow fuzzy dependency path matching in our matcher, and experiment with a rule set that is not edited manually, but acquired from running an information extraction customizer.

3 Fuzzy Dependency Path Matching

In this year's system, we allow *fuzzy* match of extraction rules with the following steps: we first extract dependency paths between name-fill pairs; we then perform fuzzy match between the extracted paths and extraction rules, using an edit-distance-based algorithm described in (He and Grishman, 2015): we split dependency path into nodes, where each node consists of a dependency label and the word it governs, and compute edit distance between node sequences. Following (He and Grishman, 2015), we use insertion cost 0.3, deletion cost 1.2, and substitution cost 0.8.

Finally, we record exact matches as relations, and record edit operations for fuzzy matches. We train an edit operation classifier to determine if the fuzzy matched dependency path warrants a relation between a name and a filler.

The features we use to train the classifier are reported in Table 1. Note that as we calculate edit-distance between two nodes on the dependency path, we are able to construct features around the dependency label, the word, and the lemma of the word. We train the classifier on the dataset collected by Angeli et al. (2014).

4 Relation Extraction Rule Construction with ICE

The lexical and dependency path rules we use in each year's KBP participation are prepared manually; they therefore represent the collective effort of computational linguistic researchers for several summers. This year, we try to find out how far an information extraction customization tool can bring us for KB construction: instead of using the rules we have collected over the years, we experimented with starting from scratch and collecting relation extraction rules with ICE.

ICE [the Integrated Customization Environment] is an information extraction customization tool, which lowers the barriers to IE system development by providing guidance while letting the user retain control, and by allowing the user to interact in terms of the words and phrases of the domain, with a minimum of formal notation. Users are able to construct new entity and relation types for a new domain, by providing one or two seed entities (for entity set expansion) or phrases (for relation extraction rule bootstrapping).

We mainly use ICE to bootstrap slot filling rules for the KBP Cold Start task. The ICE bootstrapper follows the style of Snowball (Agichtein and Gravano, 2000). For each relation, we first provide ICE with one or two seed dependency paths derived from KBP slot descriptions. The bootstrapper then searches for more dependency paths that connect the same entities as the seed paths and returns them to the user for review. The paths that are approved by the user are then sent back to ICE for the next iteration of bootstrapping. We convert dependency paths to English phrases so that they are more readable to users who are not familiar with NLP.

We conduct our bootstrapping experiment on the 2008 APW section of Gigaword. For each relation, the user reviews maximally 20 dependency paths for one iteration, and stops after 5 iterations. The seed paths derived from slot descriptions are manually added to both ICE and the Cold Start system. We do not make any change to the rule set generated by ICE.

Feature	Explanation
INS/DEL_LABEL	dependency label of inserted/deleted node
INS/DEL_WORD	word of the inserted/deleted node
INS/DEL_LEMMA	lemma of the inserted/deleted node
SUB_SAME_LABEL	whether the substituted node and the new node has the same dep. label
SUB_SAME_WORD	whether the substituted node and the new node has the same word
SUB_SAME_LEMMA	whether the substituted node and the new node has the same lemma
SUB_LABELS	conjunction of the substituted and the new dependency label
SUB_WORDS	conjunction of the substituted and the new words
SUB_LEMMA	conjunction of the substituted and the new lemmas
SUB_FROM_LABEL	dependency label of the substituted node
SUB_TO_LABEL	dependency label of the new node
SUB_FROM_WORD	substituted word
SUB_FROM_LEMMA	substituted lemma
SUB_TO_WORD	new word
SUB_TO_LEMMA	new lemma
SUB_FROM	conjunction of substituted label and word
SUB_TO	conjunction of new label and word

Table 1: Features for the fuzzy match classifier

5 Results

5.1 Fuzzy match relation extractor

We report the experimental results (hop0+hop1) on TAC 2014 Cold Start evaluation data in Table 2. Note that given the sparsity of the cold start evaluation pool, using the 2014 assessments directly (as we do) is likely to underestimate the system's true performance.

	P	R	F1
2014	0.576	0.133	0.215
2014re	0.329	0.142	0.217
Fuzzy	0.314	0.195	0.241

Table 2: Effect of fuzzy matching. 2014: KB submitted to KBP 2014; 2014re: reconstructed 2014 system with misc. improvements; Fuzzy: 2014re+fuzzy match of extraction rules

Systems reported in Table 2 utilize all available components in our pipeline, including the core tagger, the pattern tagger, and the distantly supervised tagger. Comparing *2014re*, which uses the exact pattern tagger against *Fuzzy*, which uses the fuzzy pattern tagger, we observe that the Fuzzy tagger improves 5% absolute recall at the cost of less than 2% precision, and thus improves the F1 score of the whole pipeline. We do not have final scores for the official run as we are preparing this draft, but preliminary results from the 2015 evaluation shows that the fuzzy tagger still improves the overall performance for hop0, but is penalized heavily on hop1. This is because slots with NIL response are introduced this year: the fuzzy tagger incorrectly fills several NIL slots, which leads to a lot of false positives during hop1.

5.2 ICE generated rules

	Core Tagger			with ICE rules		
	hop0	hop1	all	hop0	hop1	all
P	0.71	0.21	0.47	0.44	0.15	0.34
R	0.06	0.02	0.04	0.08	0.02	0.05
F1	0.11	0.03	0.07	0.13	0.03	0.09

Table 3: Knowledge base construction results with and without the rules collected by ICE

We report results obtained by ICE generated rules on TAC 2014 Cold Start evaluation data in Table 3. In the *Core Tagger* setting, we only use the core tagger (cf. Section 2); in the *with ICE rules* setting, we use the core tagger and a fuzzy matcher with rules acquired by ICE. We notice that the ICE-derived rules improve the recall of the core tagger in hop0, but the improvement is not significant enough to make real difference in hop1. When analyzing the rules collected by ICE, we find it is one third of the size of the NYU curated rules that we actually use in KBP evaluations.

We suspect that this lack of coverage comes in part from the choice of the data. We bootstrap from only one year of AP news articles, which have consistent styles and do not provide enough diversity of expressions.

6 Related Work

Various strands of recent research on reducing annotation effort for NLP and information extraction motivated our experiments. The first strand is active learning for relation extraction. Sun and Grishman (2012) used local and global data views to select annotation instances and was able to build a relation extractor with small annotation cost. This work was later extended by Fu and Grishman (2013). Our relation bootstrapper and entity set expander can be considered as simplified versions of these systems. Angeli et al. (2014) applied active learning techniques to improve distant supervision for knowledge base construction. They also acquired label annotation via crowd sourcing. Our setting is different in that it does not crowd-source user data and does not assume a pre-existing collection of relation triples.

Another strand focuses on building straightforward and easily interpretable models for relation and event extraction, which our KBP system and the ICE relation extraction customizer try to follow. Specifically, Valenzuela-Escárcega et al. (2015) presented a pattern language and a system to extract events, while Bronstein et al. (2015) detects event triggers by measuring the similarity between candidates and trigger terms mentioned in annotation guidelines.

Finally, ICE (He and Grishman, 2015) is developed among several tools to support rapid construction of user-defined information extractors: e.g. the WIZIE system from IBM Research (Li et al., 2012), the SPIED system (Gupta and Manning, 2014) from Stanford, and PROPMINER system from T. U. Berlin (Akbik et al., 2013). See (He and Grishman, 2015) for a discussion on these systems.

7 Conclusion and Future Work

We described experiments we conducted around the 2015 KBP Cold Start task, in particular our experiments on using ICE to facilitate knowledge base construction. Initial results showed that ICE collects rules that slightly helped knowledge base construction, and we plan to continue working on ICE, as more needs to be done if we want ICE to produce rules that perform close to human curated ones.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 85–94.

Alan Akbik, Oresti Konomi, Michail Melnikov, et al. 2013. Propminer: A workflow for interactive information extraction and exploration using dependency trees. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: Systems Demonstrations*, pages 157–162.

Gabor Angeli, Julie Tibshirani, Jean Y. Wu, and Christopher D. Manning. 2014. Combining distant and partial supervision for relation extraction. In *EMNLP*.

Ofer Bronstein, Ido Dagan, Qi Li, Heng Ji, and Anette Frank. 2015. Seed-based event trigger labeling: How far can event descriptions get us? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 372–376, Beijing, China.

Lisheng Fu and Ralph Grishman. 2013. An efficient active learning framework for new relation types. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 692–698, Nagoya, Japan.

Sonal Gupta and Christopher Manning. 2014. SPIED: Stanford pattern based information extraction and diagnostics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, pages 38–44.

Yifan He and Ralph Grishman. 2015. ICE: Rapid information extraction customization for nlp novices. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 31–35, Denver, CO. Association for Computational Linguistics.

Yunyao Li, Laura Chiticariu, Huahai Yang, Frederick R Reiss, and Arnaldo Carreno-Fuentes. 2012. WizIE: a best practices guided development environment for information extraction. In *Proceedings of the ACL 2012 System Demonstrations*, pages 109–114.

Thien Huu Nguyen, Yifan He, Maria Pershina, Xiang Li, and Ralph Grishman. 2014. New york university 2014 knowledge base population systems. In *Proc. Text Analysis Conference (TAC2014)*.

Ang Sun and Ralph Grishman. 2012. Active learning for relation type extension with local and global data views. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1105–1112.

Ang Sun, Ralph Grishman, Wei Xu, and Bonan Min. 2011. New york university 2011 system for kbp slot filling. In *Proceedings of the Text Analytics Conference*.

Marco A. Valenzuela-Escárcega, Gustave Hahn-Powell, Thomas Hicks, and Mihai Surdeanu. 2015. A domain-independent rule-based framework for event extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Assian Federation of Natural Language Processing: Software Demonstrations (ACL-IJCNLP)*.