Pocket Knowledge Base Population

Travis Wolfe **Mark Dredze** **Benjamin Van Durme**
Human Language Technology Center of Excellence
Johns Hopkins University

Abstract

Existing Knowledge Base Population methods extract relations from a closed relational schema with limited coverage, leading to sparse KBs. We propose *Pocket Knowledge Base Population (PKBP)*, the task of dynamically constructing a KB of entities related to a query and finding the best characterization of relationships between entities. We describe novel Open Information Extraction methods which leverage the PKB to find informative trigger words. We evaluate using existing KBP shared-task data as well as new annotations collected for this work. Our methods produce high quality KBs from just text with many more entities and relationships than existing KBP systems.

1 Introduction

Much of human knowledge is contained in text in books, encyclopedias, the internet, and written communications. Building knowledge bases to store, search, and reason over this information is an important problem in natural language understanding. A lot of work in knowledge base population (KBP) has focused on the NIST Text Analysis Conference track of the same name, and specifically the slot filling task. Slot Filling (SF) defines a relational schema similar to Wikipedia infoboxes. SF KBP systems extract facts from text corresponding to an entity called the query.

This work addresses two issues concerning SF KBP. First, the SF schema has strict semantics for the relations which can be extracted, and thus no SF relation can be extracted for most related entities, leading to sparse KBs. Second, because SF has a small static schema, most research has focused on batch processing for a single schema, limiting downstream usefulness. This means KBs built by slot filling have limited applicability in some real world settings of interest.

We address these issues by proposing *Pocket Knowledge Base Population*. Pocket KBs (PKBs) are dense entity-centric KBs dynamically constructed for a query. In both SF and pocket KBP, a query is an entity of interest and a document mentioning that entity. However, in PKB the primary goal is to populate the KB with nodes for all entities related to the query, irrespective of any prior beliefs about relations. PKB edges store representations of mentions referring to the entities connected by that edge, and thus may better serve downstream tasks which don't perfectly align to a particular schema.

We describe a PKBP system which builds KBs from text corpora. This includes unsupervised methods for finding related entities and mentions of them and the query with accuracies of 89.5 and 93.1 respectively when evaluated on SF queries. We also propose novel entity-centric Open IE (Banko et al., 2007) methods for characterizing the relationship between entities which perform twice as well as a syntactically-informed baseline. Our contributions also include a comparison between pocket and SF KBs constructed on SF queries, showing our KBs are multiple times larger while remaining high quality. We make our system publicly available.[1]

2 Pocket Knowledge Base Population

The defining characteristic of pocket KBs is they are small, entity-centric, and dynamically generated according to a query. Most work in KBP is centered around batch processing with a relation extractor, whereas PKBP is based on entity men-

[1] https://hub.docker.com/r/hltcoe/pocket-knowledge-base-population

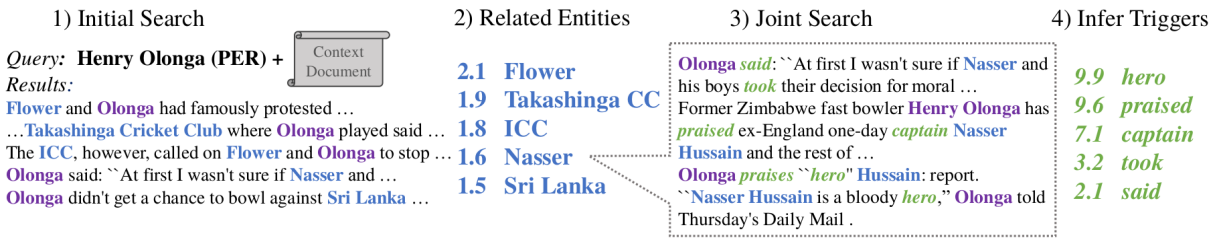

Figure 1: High level steps of the PKB construction process. Example PKBs can be found in Table 3. The first two steps of initial search and related entities are described in §3.1, the third step of joint searching in §3.2, and finally extracting triggers in §3.3.

tion search and ad-hoc trigger extraction. PKBs resemble a hub and spoke graph where the hub is the query and nodes on the outside are related entities. The spokes represent mentions proving that an entity is related to the query. Traversing this graph by crossing a spoke is akin to building a new PKB with that related entity as the new hub.

PKBs are aids for "knowledge-based" search over a document collection. KBs are useful for question answering (Yao, 2014) and PKBs serve this goal because related entities are good answer candidates for questions about a query. SF KBs are specialized to the case where you know the questions ahead of time, like "Where does Mary *work*?" and "Who is ACME's *parent company*?". PKBs offer a set of answers for queries with clues as to their relationship, like Marc Bolland and Stuart Rose are related because of events described using the word *replaced*. KBs are also useful information retrieval (IR) tools (Dietz and Schuhmacher, 2015) for human guided corpus exploration. PKBs serve this goal by providing ranked lists of related entities and over the mentions describing their relationship to the query.

We describe a system which achieves the goals of PKBP using low-resource and unsupervised methods. We discuss how PKBs are built in §3 and evaluate their quality in §4 on SF queries.

3 Construction

There are three steps to PKB construction: §3.1 discovering related entities, §3.2 finding mentions of the query and related entities, and §3.3 extracting trigger word explanations.

3.1 Discovering Related Entities

Candidate related entities are collected by searching for mentions of the query and taking other mentions which appear in the results. This process is similar to cross document entity coreference and we adopt the vector space model (Bagga and Baldwin, 1998). First, **triage features** are used to locate sentences in an inverted index, including the mention headword and all word unigrams and bigrams (case sensitive and insensitive).

Next, we use **context features**: a word unigram tf-idf vector. We implement a compromise between Cucerzan (2007) (used sentences before, after, and containing a mention) and Bagga and Baldwin (1998) (used any sentence in a coreference chain). Instead of running a coreference resolver, we use the high-precision heuristic of linking mentions with the same headword and NER type. Terms are weighted as $\frac{2}{1+d}$ where d is the distance in sentences to the nearest mention.

Our **attribute features** are a generalization of Mann and Yarowsky (2003). They train a few ad-hoc relation extractors like `birth_year` and `occupation` from seed facts. Their extractions provide high-precision signal for merging entity mentions. We found extracting all `NNP`$\star$ or capitalized `JJ`$\star$ words within 4 edges in a dependency tree was less sparse, requires no seeds, and produced similar quality attributes. We union these attributes across mentions found by the headword and NER type coreference heuristic to build a fine-grain tf-idf vector. We use the same $\frac{2}{1+d}$ re-weighting for attributes, except where d is the distance in dependency edges to the entity mention head. The closest attributes are descriptors within a noun phrase like `HEAD-nn-Dr.`. We include the NER type of the headword to distinguish between attributes like `PERSON-nn-American` and `ORGANIZATION-nn-American`.

Given the triage features $t(m)$, context features $c(m)$, and attribute features $a(m)$, we search for mentions m which maximize

$$(1 + \alpha_t \cos\theta_t)(1 + \alpha_c \cos\theta_c)(1 + \alpha_a \theta_a)$$

where $\cos\theta_t$ is the cosine similarity between

$t(m_{query})$ and $t(m)$. We only consider the subset of mentions that have $\cos\theta_t > 0$, which can be efficiently retrieved via an inverted index.

Any mention with a score higher than τ is considered coreferent with the query. We extract mentions in the same sentences as the query as candidate related entities if they have an NER type of `PER`, `ORG`, or `LOC`. We link candidate mentions against entities in the PKB using the same coreference score used to retrieve query mentions. If a candidate's best link has a score $s < \tau$, we promote it to an entity and add it to the PKB with probability $1 - \frac{s}{\tau}$.[2]

3.2 Joint Linking of Related Entities

At this point there are on the order of 100 mentions of the query and 20 to 50 related entities.[3] For each entity, we perform a joint search for it and the query. These entity co-occurrences will form the spokes in the PKB and be used to characterize the relationship between and relatedness to the query.

Joint entity searches are similar to single-mention searches in §3.1 with two differences. First, instead of having a single mention to compute feature vectors from, there are multiple. Feature vectors for entities are built up from mentions, where the weight of a mention $w(m) = \rho^b$ for $\rho \in (0, 1)$ and b is how many mentions were linked before m. Second, we are scoring mention pairs (with both mentions in the same sentence) as the geometric mean of the coreference scores of both links. The coreference score function does not need to change, but the triage step does: we only consider sentences which have $\cos\theta_t > 0$ for both the query and the related entity and use the same τ. Entity relatedness is a function of how often entities are mentioned together. We modeled it as the sum of the joint entity linking probabilities, where the probability of a link is $\text{logit}^{-1}(\frac{s}{\tau})$.

3.3 Trigger Word Analysis

At this stage we have found on the order of 2 to 20 sentences which mention the query and a related entity which will be used to determine the relation between them. There is work on rule-based (Banko et al., 2007; Fader et al., 2011; Angeli et al., 2015), supervised (Mausam et al., 2012), and distantly-supervised (Mintz et al., 2009) methods for characterizing relations in text. Our method is similar to distant supervision, where a KB of known facts is used to infer how relations are expressed, but we use supervision from the KB being constructed. We cast the problem of characterizing a relation as a search for *trigger words*. We state our priors on trigger words and condition on the data to find likely triggers.

Predicate (triggers) and arguments are syntactically close together. Assuming the related entity mention heads are arguments, we compute the probability that these two random walks in a dependency tree end up at the same token. This serves as a weak syntactically informed prior.

Information is conveyed as a surprisal under a background distribution (codebook). We compute a unigram distribution over words which are likely under our syntactic prior for triggers (conditioned on the NER type of the two arguments). We use this marginal distribution as a codebook. We divide out this codebook probability in every pair of related entity mentions in the PKB giving a cost in bits (log probability ratio) of each trigger word.

Repetition indicates importance. We sum the costs for each trigger across sentences. We weaken this assumption by averaging the max and the sum for each trigger for the final score.

This process yields a score for every trigger word, and we use the top k triggers to characterize the relationship between entities. For each trigger we keep, we also maintain provenance information for mentions using a given trigger.

4 Experiments

We use the TAC SF13 query entities to evaluate our methods; 50 person and 50 organization entities are used as queries to construct 100 PKBs. 70 of the 100 query entities were NIL (26/60 PER and 44/50 ORG), meaning that they do not appear in the TAC KB, though our methods aren't in principle sensitive to this because they create entities on the fly. We use annotated versions of Gigaword 5 (Parker et al., 2011; Ferraro et al., 2014) and English Wikipedia (February 24, 2016 dump) to construct our PKBs.[4] We use Amazon Mechanical Turk workers as annotators. We generated our PKBs with $\tau = 15$, $\rho = 0.5$, $\alpha_t = 40$, $\alpha_c = 20$, and $\alpha_a = 10$. These constants were tuned by hand

[2]Mentions with a score near τ may be coreferent, so we prefer low scoring mentions to avoid over-splitting entities.

[3]These values depend on the query (which are more or less rare in a corpus) and pruning thresholds (for our experiments we stop at 100 query mentions)

[4]We do not use the coreference annotations provided by Annotated Gigaword, only the features described in §3.1.

and are not sensitive to small changes. We take a subset of the PKB which covers the 15 most related entities and the one-best trigger for each. We call these "explanations" where each is a sentence with three labels: a) a mention of the query m_q, b) a mention of the a related entity m_r, and c) a trigger word t.

Entity Linking and Relatedness For each explanation, we ask: COREF: Does the query mention refer to the same entity as m_q? RELATED: Is the query entity meaningfully related to the referent of m_r? These annotations are not done by the same annotators to avoid confirmation bias. Worried annotators might be lulled into thinking all COREF instances were true, we made the task ternary by adding an intruder entity (randomly drawn from SF13 queries). Annotators were shown m_q and could choose coreference with the query, the intruder, or neither.[5] We drop annotations from annotators who chose an intruder[6] because we know these to be incorrect, and compute accuracy as proportion of the remaining annotations which chose the query.

RELATED was posed as a binary task of whether m_r is more related to the query or the intruder (without highlighting m_q). In positive cases, the annotator should observe that sentence shown contains a mention of the query entity and explains why they are related. The results are in Table 1.

Our system retrieves coreferent and related mentions with high accuracy. For coreference, mistakes usually happen when there is significant lexical overlap but some distinguishing feature that proves too subtle for our system to doubt the match, like Midwest High Speed Rail Association vs U.S. High Speed Rail Association or [English] Nationwide Building Society vs Irish Nationwide Building Society.

For relatedness, the biggest source of errors are news organizations listed as related entities because it is common to see sentences like *"**Mohammed Sobeih**, Moussa's deputy, told **The Associated Press** on Monday that..."*. Future work might address this problem by using normalized measures of statistical relatedness like PMI rather than raw co-occurrence counts.

Trigger Words To evaluate the informativeness of chosen triggers, we present annotators with m_q, m_r, and two potential trigger words highlighted. One trigger is chosen according to §3.3 and the other is an NN⋆ | VB⋆ | JJ⋆ | RB⋆ word in the projection of the dependency node dominating both entities.[7] The annotator may choose either trigger as a good characterization of the situation involving m_q and m_r, or label neither as sufficient. Note that this baseline is strong: it shares the entity linking (§3.2), trigger sentence selection (§3.3), and dependency parse tree as our system. We report the results in Table 2.

	PER	ORG	All
COREF	94.6	91.5	93.1
RELATED	90.7	88.2	89.5
COREF and RELATED	86.6	80.9	83.9

Table 1: PKB entity accuracy.

	System	Intruder	Neither
Person	29.4	12.4	58.2
Organization	29.1	17.3	53.7
All	29.2	14.7	56.1

Table 2: Related entity trigger identification.

Our method is chosen about twice as often as a syntactically informed baseline, but fails to find a high quality trigger word more than half of the time. Some mistakes are caused by rare but oft-repeated words like "50" in: *"**Bolland**, **50**, ... will replace Briton **Stuart Rose**"*. "50" has nothing to do with the relationship between Bolland and Rose, but it's repeated in 4 sentences about both of them, a stylistic coincidence our system cannot ignore. In other cases there is no word in situ which can explain entities' relatedness, like *"... the day after **Wimbledon concludes**, **Montcourt** must serve a five-week ban and ..."*. The author and the reader can likely infer that Montcourt *competed* at Wimbledon, but this fact is not explicitly committed to, limiting our systems ability to extract a trigger.

Related Entities vs Slot Fillers There is no fair way to evaluate systems without a common schema, but we offer some extraction statistics. On SF13 queries our system generated 17.6 relevant entities/query,[8] each having 4.6 trigger words/pair, 2.1 mentions/trigger word, and 9.8

[5] The order of the intruder and the query were randomized.

[6] This affected 6.1% of COREF annotations.

[7] If no nodes match this, we walk up the tree until we find a node which has at least one allowed descendant.

[8] This is given a cap of 20 relevant entities per query to avoid a skewed average and keep construction time down.

Query Entity		Related Entity		Triggers
Marc Bolland	PER	Dalton Philips	PER	*appointed, departure, following, move*
Marc Bolland	PER	Stuart Rose	PER	*replace, 50, Briton*
Marc Bolland	PER	Marks & Spencer	ORG	*departure, CEO, become, following*
Henry Olonga	PER	Givemore Makoni	PER	*club, president, done, played*
Henry Olonga	PER	England	LOC	*cricketer, asylum, hiding, quit*
Henry Olonga	PER	Harare	LOC	*hiding, armbands, wore*
Mohammad Oudeh	PER	Munich	LOC	*massacre, briefed, defended*
Mohammad Oudeh	PER	Fatah Revolutionary Council	ORG	*faction, belonged, return*
Mohammad Oudeh	PER	Gaza Strip	LOC	*allows, asked, host*
A123 Systems LLC	ORG	Fisker	ORG	*supplier, struck, recall, owns*
A123 Systems LLC	ORG	Watertown, Massachusetts	LOC	*produces, batteries, company*
A123 Systems LLC	ORG	Obama	PER	*plant, opening, Granholm*
United Steelworkers of America	ORG	Curt Brown	PER	*spokesman, rejected, contracts*
United Steelworkers of America	ORG	Wayne Fraser	PER	*negotiator, spokesman, union*
United Steelworkers of America	ORG	Jerry Fallos	PER	*boss, broke, shut, local*
BNSF	ORG	Santa Fe	LOC	*asked, vote*
BNSF	ORG	Chapman	ORG	*venture, help, transition, joint*
BNSF	ORG	Robert Krebs	PER	*Burlington, chairman*

Table 3: Examples of slices of PKBs for the three most related entities for six queries and the best triggers for each pair. Supporting sentences for related entities and trigger words are not shown.

mentions/pair. In extractions from *all* systems in the SF13 evaluation (pooling answers, filtering out incorrect), they filled 6.0 slots/query with 14.2 fillers/query and 38.3 mentions/query as provenance. Some slots have string-valued fillers, but many could be related entities in the PKB sense. In these cases, we found 2.2 entities/query overlapping, 1.7 fillers not in their corresponding PKB and 10.8 related entities which weren't fillers.

5 Related Work

Blanco and Zaragoza (2010) study the information retrieval problem of finding *support sentences* which explain the relationship between a query and an entity, which is similar to this work. Our work addresses two new aspects of this problem: 1) how to automatically find related entities, which are assumed given in that work and 2) how to find the salient parts of support sentences (trigger words) by aggregating evidence across sentences.

This work shares goals with Dalton and Dietz (2013) and Dietz and Schuhmacher (2015), who create "knowledge sketches": distributions over documents, entities, and relations related to a query. The primary difference is that our work creates a KB instead of returning results from an existing one. They use Freebase for relations and Wikipedia for anchor text and links. Our approach uses parsed and NER tagged text.

Open vocabulary characterization of entities was investigated by Raghavan et al. (2004). They found intersecting entity language models yields common descriptors. Their notion of similarity (e.g. Ronald Reagan and Richard Nixon are both *presidents*) is different from our notion of relatedness (e.g. Alexander Haig and Princeton, NJ are related via *Meredith* – Haig's sister).

Finally other work has used Open IE for SF KBP. Soderland et al. (2013) and Finin et al. (2015) manually created a mapping between the Ollie (Mausam et al., 2012) and SF schemas. Angeli et al. (2015) perform OpenIE and then map between their schema and SF with PMI^2.

6 Conclusion

We propose *Pocket Knowledge Base Population* for dynamically building dense entity-centric KBs. We evaluate our methods on SF queries and find high accuracies of related entity discovery and coreference. We propose novel Open Information Extraction methods which leverage the PKB to identify trigger words and show they are effective at explaining related entities. In future work we hope to use PKBs for tasks like QA and IR.

Acknowledgments

This research was supported by the Human Language Technology Center of Excellence (HLTCOE) and Bloomberg L.P. The views and conclusions contained in this publication are those of the authors.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 344–354. http://www.aclweb.org/anthology/P15-1034.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '98, pages 79–85. https://doi.org/10.3115/980845.980859.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI'07, pages 2670–2676. http://dl.acm.org/citation.cfm?id=1625275.1625705.

Roi Blanco and Hugo Zaragoza. 2010. Finding support sentences for entities. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, SIGIR '10, pages 339–346. https://doi.org/10.1145/1835449.1835507.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics, Prague, Czech Republic, pages 708–716. http://www.aclweb.org/anthology/D/D07/D07-1074.

Jeffrey Dalton and Laura Dietz. 2013. Constructing query-specific knowledge bases. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*. ACM, New York, NY, USA, AKBC '13, pages 55–60. https://doi.org/10.1145/2509558.2509568.

Laura Dietz and Michael Schuhmacher. 2015. An interface sketch for queripidia: Query-driven knowledge portfolios from the web. In Krisztian Balog, Jeffrey Dalton, Antoine Doucet, and Yusra Ibrahim, editors, *Proceedings of the Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval, ESAIR 2015, Melbourne, Australia, October 23, 2015*. ACM, pages 43–46. https://doi.org/10.1145/2810133.2810145.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '11, pages 1535–1545. http://dl.acm.org/citation.cfm?id=2145432.2145596.

Francis Ferraro, Max Thomas, Matthew R. Gormley, Travis Wolfe, Craig Harman, and Benjamin Van Durme. 2014. Concretely annotated corpora. In *The NIPS 2014 AKBC Workshop*.

Tim Finin, Dawn Lawrie, Paul McNamee, James Mayfield, Douglas Oard, Nanyun Peng, Ning Gao, Yiu-Chang Lin, Josh MacLin, and Tim Dowd. 2015. Hltcoe participation in tac kbp 2015: Cold start and tedl. In *Text Analytics Conference (TAC)*.

Gideon S. Mann and David Yarowsky. 2003. Unsupervised personal name disambiguation. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*. Association for Computational Linguistics, Stroudsburg, PA, USA, CONLL '03, pages 33–40. https://doi.org/10.3115/1119176.1119181.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP-CoNLL '12, pages 523–534. http://dl.acm.org/citation.cfm?id=2390948.2391009.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, pages 1003–1011.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition, linguistic data consortium. Technical report, Technical report, Technical Report. Linguistic Data Consortium, Philadelphia.

Hema Raghavan, James Allan, and Andrew McCallum. 2004. An exploration of entity models, collective classification and relation description .

Stephen Soderland, John Gilmer, Robert Bart, Oren Etzioni, and Daniel S Weld. 2013. Open information extraction to kbp relations in 3 hours. In *TAC*.

Xuchen Yao. 2014. *Feature-driven Question Answering with Natural Language Alignment*. Ph.D. thesis.