Joint Bootstrapping of Corpus Annotations and Entity Types

Hrushikesh Mohapatra Siddhanth Jain Soumen Chakrabarti*
IIT Bombay

Abstract

Web search can be enhanced in powerful ways if token spans in Web text are annotated with disambiguated entities from large catalogs like Freebase. Entity annotators need to be trained on sample mention snippets. Wikipedia entities and annotated pages offer high-quality labeled data for training and evaluation. Unfortunately, Wikipedia features only one-ninth the number of entities as Freebase, and these are a highly biased sample of well-connected, frequently mentioned "head" entities. To bring hope to "tail" entities, we broaden our goal to a second task: assigning types to entities in Freebase but not Wikipedia. The two tasks are synergistic: knowing the types of unfamiliar entities helps disambiguate mentions, and words in mention contexts help assign types to entities. We present TMI, a bipartite graphical model for joint type-mention inference. TMI attempts no schema integration or entity resolution, but exploits the above-mentioned synergy. In experiments involving 780,000 people in Wikipedia, 2.3 million people in Freebase, 700 million Web pages, and over 20 professional editors, TMI shows considerable annotation accuracy improvement (e.g., 70%) compared to baselines (e.g., 46%), especially for "tail" and emerging entities. We also compare with Google's recent annotations of the same corpus with Freebase entities, and report considerable improvements within the people domain.

1 Introduction

Thanks to automatic information extraction and semantic Web efforts, keyword search over unstructured Web text is rapidly evolving toward entity- and type-oriented queries (Guo et al., 2009; Pantel et al., 2012) over semi-structured databases such as Wikipedia, Freebase, and other forms of Linked Data.

A key enabling component for such enhanced search capability is a type and entity *catalog*. This includes a directed acyclic graph of types under the *subTypeOf* relation between types, and entities attached to one or more types via *instanceOf* edges. YAGO (Suchanek et al., 2007) provides such a catalog by unifying Wikipedia and WordNet, followed by some cleanup.

Another enabling component is an *annotated corpus* in which token spans (e.g., the word "Albert") are identified as a mention of an entity (e.g., the Physicist Einstein). Equipped with suitable indices, a catalog and an annotated corpus let us find "scientists who played some musical instrument", and answer many other powerful classes of queries (Li et al., 2010; Sawant and Chakrabarti, 2013).

Consequently, accurate corpus annotation has been intensely investigated (Mihalcea and Csomai, 2007; Cucerzan, 2007; Milne and Witten, 2008; Kulkarni et al., 2009; Han et al., 2011; Ratinov et al., 2011; Hoffart et al., 2011). With two exceptions (Zheng et al., 2012; Gabrilovich et al., 2013) that we discuss later, public-domain corpus annotation work has almost exclusively used Wikipedia and derivatives, partly because Wikipedia provides not only a standardized space of entities, but also reliably labeled mention text within its own documents, which can be used to train machine learning algorithms for entity disambiguation.

However, the high quality of Wikipedia comes at the cost of low entity coverage (4.2 million) and bias toward often-mentioned, richly-connected "head" entities. Hereafter, Wikipedia entities are called W. Freebase has fewer editorial controls, but has at least nine times as many entities. This is particularly perceptible for *people* entities: one needs to be relatively famous to be featured on Wikipedia, but Freebase is less selective. Hereafter, Freebase entities are called F.

As in any heavy-tailed distribution, even relatively obscure entities from $F \setminus W$ are *collectively* mentioned a great many times on the Web, and including them in Web annotation is critical, if entity-oriented search is to impact the vast number of tail

*soumen@cse.iitb.ac.in

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 436–446,
Seattle, Washington, USA, 18-21 October 2013.

queries submitted to Web search engines.

Primary goal — corpus annotation: We have thus established a pressing need to bootstrap from a small entity catalog W (such as Wikipedia entities), and a small *reference* corpus C_W (e.g., Wikipedia text) reliably annotated with entities from W, to a much larger catalog F (e.g., Freebase), and an open-domain large *payload* corpus C (e.g., the Web).

We can and will use entities in $F \cap W$[1] in the bootstrapping process, but the real challenge is to annotate C with mentions m of entities in $F \setminus W$. Unlike for $F \cap W$, we have no training mentions for $F \setminus W$. Therefore, the main disambiguation signal is from the immediate entity neighborhood $N(e)$ of the candidate entity e in the Freebase graph. I.e., if m also reliably mentions some entity in $N(e)$, then e becomes a stronger candidate. Unfortunately, for many "tail" entities $e \in F \setminus W$, $N(e)$ is sparse. Is there hope for annotating the Web with tail entities? Here, we achieve enhanced accuracy for the primary annotation goal by extending it with a related secondary goal.

Secondary goal — entity typing: If we had available a suitable type catalog $\mathcal{T}$ with associated entities in W, which in turn have known textual mentions, we can build models of contexts referring to types like chemists, sports people and politicans. When faced with people called John Williams in $F \setminus W$, we may first try to associate them with types. This can then help disambiguate mentions to specific instances of John Williams in $F \setminus W$. In principle, useful information may also flow in the reverse direction: words in mention contexts may help assign types to entities in $F \setminus W$. For reasons to be made clear, we choose YAGO (Suchanek et al., 2007) as the type catalog $\mathcal{T}$ accompanying entities in W.

Our contributions: We present TMI, a bootstrapping system for solving the two tasks jointly. Apart from matches between the context of m and entity names in $N(e)$, TMI combines and balances evidence from two other sources to decide if e is mentioned at token span m, and has type t:

- a language model for the context in which entities of type t are usually mentioned
- correlations between t and certain path features generated from $N(e)$.

TMI uses a novel probabilistic graphical model formulation to integrate these signals. We give a detailed account of our design of node and edge potentials, and a natural reject option (recall/precision tradeoff).

We report on extensive experiments using YAGO types, Wikipedia entities and text, Freebase entities, and text from ClueWeb12[2], a 700-million-page Web corpus. We focus on all people entities in Wikipedia and Freebase, and provide three kinds of evaluation. First, we evaluate TMI on over 1100 entities in $F \cap W$ and 5500 snippets from Wikipedia text, where it visibly improves upon baselines and a recently proposed alternative method (Zheng et al., 2012). Second, we resort to extensive manual evaluation of annotation on ClueWeb12 Web text with Freebase entities, by professional editors at a commercial search company. TMI again clearly outperforms strong baselines, doing particularly well for nascent or tail entities. TMI improves per-snippet accuracy, for some classes of entities, from 46% to 70%, and pooled F1 score from 66% to 73%. Third, we compare TMI annotations with Google's FACC1 (Gabrilovich et al., 2013) annotations restricted to people; TMI is significantly better. Our annotations and related data can be downloaded from `http://www.cse.iitb.ac.in/~soumen/doc/CSAW/`. To our knowledge, this is among the first reports on extensive human evaluation of machine annotation for $F \setminus W$ on a large Web corpus.

2 Related work

The vast majority of entity annotation work (Mihalcea and Csomai, 2007; Cucerzan, 2007; Milne and Witten, 2008; Kulkarni et al., 2009; Han et al., 2011; Ratinov et al., 2011; Hoffart et al., 2011) use Wikipedia or derivative knowledge bases. (Ritter et al., 2011) and (Zheng et al., 2012) are notable exceptions. (Ritter et al., 2011) use entity names for distant supervision in POS tagging, chunking and broad named entity typing in short tweets, which are different from our goals.

Recently, others have investigated inferring types

[1] With F=Freebase and W=Wikipedia, $F \cap W \approx W$ but not quite; $W \setminus F$ is small but non-empty.

[2] `http://lemurproject.org/clueweb12/`

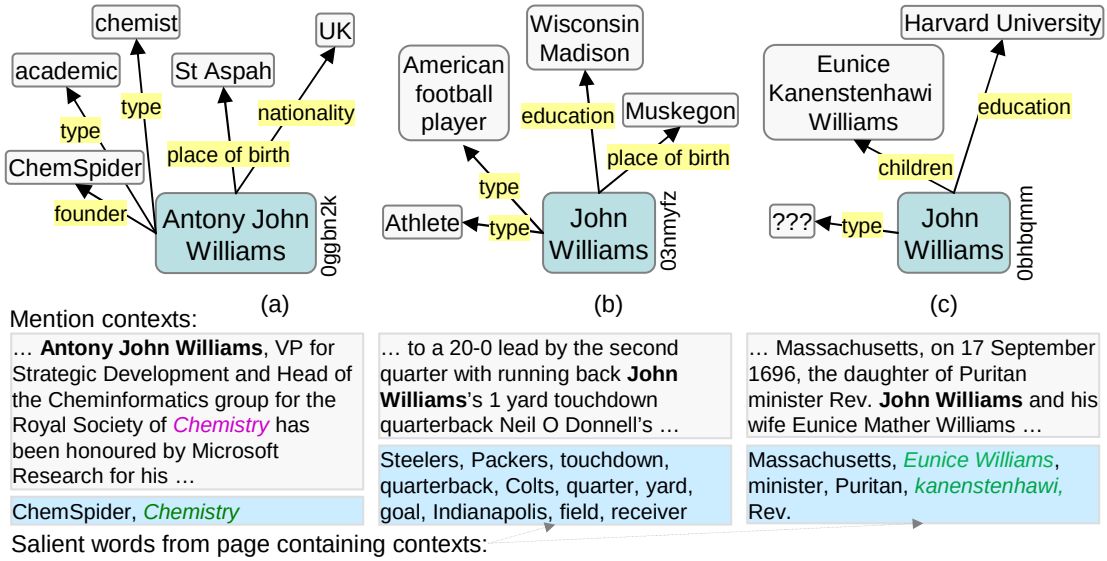

Fig. 1: Signal synergies. Three of the many people mentioned as "John Williams" on the Web are shown, with Freebase MIDs. (a) Easy case where Freebase neighbors of 0ggbn2k match snippet and salient text, and type links are also available. (b) No match exists between Freebase neighborhood and snippet, but type links help attach the snippet to 03nmvfz. (c) Freebase provides no types, but we can provide types from YAGO based on snippet and salient text, which also match neighbors of 0bhbqmm.

of emerging entities (related to our secondary goal). In concurrent work, (Nakashole et al., 2013) propose integer linear program formulations for inferring types of emerging entities from the way their mentions are embedded in curated relation-revealing phrases. (Lin et al., 2012) earlier approached the problem using weight propagation in a bipartite graph connecting unknown to known entities via textual relation patterns. Both note that this can boost mention disambiguation accuracy.

The closest work to ours is by (Zheng et al., 2012): they use semisupervised learning to annotate a corpus with Freebase entities. Like (Milne and Witten, 2008), they depend on unambiguous entity-mention pairs to bootstrap a classifier, then apply it to unlabeled, ambiguous mentions, creating more training data. They use a per-type language model like us (§3.3), but this is used as a secondary cause for (word) feature generation, supplementing and smoothing entity-specific language models. In contrast, we use a rigorous graphical model to combine new signals, not depending on naturally unambiguous mentions. Finally, in the interest of fully automated evaluation, they limit their experiments to $F \cap W$ and Wikipedia corpus, thus differing critically from our human evaluation on F and a Web corpus.

(Gabrilovich et al., 2013) have recently released FACC1: annotations of ClueWeb09 and ClueWeb12 with Freebase entities. Their algorithm is not yet public. They report: "Due to the sheer size of the data, it was not possible to verify all the automatic annotations manually. Based on a small-scale human evaluation, the precision ... is believed to be around 80–85%. Estimating the recall is of course difficult; however, it is believed to be around 70-85%." In §5, we will see that, for people entities, TMI greatly increases recall beyond FACC1, keeping precision unimpaired.

3 The three signals

Fig. 1 shows three Freebase entities mentioned as "John Williams" in Web text, represented as nodes with Freebase "MID"s e = 0ggbn2k, 03nmvfz, and 0bhbqmm, embedded in their Freebase graph neighborhoods. Owing to larger size and higher flux, Freebase shows less editorial uniformity than Wikipedia. This shows up in missing or non-standard relation edges. Unlike YAGO, where each entity is attached to one or more types, e = 0bhbqmm does not have a type link. Many people have a link labeled *profession*, which is a second

kind of type link. Entities like e = 0bhbqmm also have small, uninformative graph neighborhoods.

Also shown are three mention contexts, each represented by the snippet immediately surrounding the mention, and salient words from the documents containing each snippet. (Salient words may be extracted as words that contribute the largest components to the document represented as a TFIDF vector.) (a) shows a favorable but relatively rare case where e = 0ggbn2k has reliable type links. We cannot assume there will be a 1-to-1 correspondence between Freebase and YAGO types, but in §3.2 we will describe how to learn associations between Freebase paths around entities and their YAGO types. The snippet and salient words show reasonable overlap with $N(e)$. In §3.4 we will describe features that characterize such overlap. In (b), e = 03nmvfz is reliably typed, but there is no direct match between $N(e)$ and the snippet. Nevertheless, the snippet can be reliably annotated with 03nmvfz if we can learn associations between types *American football player* and *Athlete* (or their approximate YAGO target types) and several context/salient words (see §3.3). In (c), e = 0bhbqmm is not reliably typed. However, there are matches between $N(e)$ and context/salient words. Once the snippet-to-entity association is established, it is easier to assign 0bhbqmm to suitable types in YAGO.

In this section we will first describe our design of the target type space, and then the tree signals that will be used in our joint inference.

3.1 Designing the target type space

By typing two people called John Williams as actor and footballer, we may also disambiguate their mentions accurately. Therefore, we need a well-organized type space where the types

- collectively cover most entities of interest,
- offer reasonable type prediction accuracy, and
- can be selected algorithmically, for any domain.

Wikipedia and Freebase have many obscure types like "people born in 1937" or "artists from Ontario" which satisfy none of the above requirements. YAGO, on the other hand, has a clean hierarchy of over 200,000 types with broad coverage and fine differentiation. Most entities in $F \setminus W$ can be accurately attached to one or more YAGO types.

if t has $< N_{\text{low}} = 5000$ member entities **then**
  reject t from our type space
  return
if t has $> N_{\text{high}} = 25000$ member entities **then**
  for each immediate child $t' \subset t$ **do**
    call $\mathbf{ChooseTypes}(t')$
else
  accept t into our type space (but do not recurse)

Fig. 2: Procedure $\mathbf{ChooseTypes}(t)$.

YAGO lists around 37,000 subtypes of *person*. To satisfy the three requirements above, we called $\mathbf{ChooseTypes}(\textit{person})$ (Fig. 2); this resulted in 130 suitable types being selected. These directly covered 80% of Freebase people; the rest could mostly be attached to slightly over-generic types within our selection.

3.2 Predicting types from entity neighborhood

There will generally not be a simple mapping between Freebase and target types. E.g., entity e may be known as a *Mayor* in Freebase, but the closest YAGO type may be *Politician*. Edge and node labels from the Freebase graph neighborhood $N(e)$ can embed many clues for assigning e a target type. E.g., e = 03nmvfz may have an edge labeled *playedFor* to a node representing the Wikipedia entity `Pittsburgh_Steelers`, which has a type link to *NFL team*. This two-hop link label sequence would repeat for a large number of players, and can be used as a feature in a classifier.

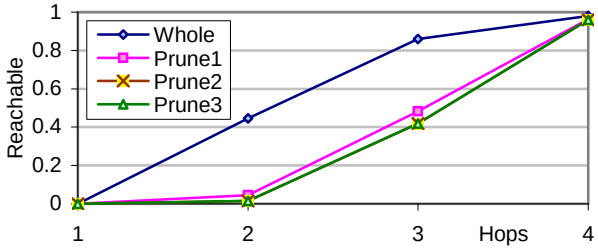

Fig. 3: Freebase has small diameter despite graph thinning. Prune1 removes paths from the whole Freebase graph that pass through nodes */user/root* and */common/topic*, Prune2 also removes node */people/person*, and Prune3 removes several other high degree hubs.

Two further refinements are needed to make this work. First, we have to collect path labels around negative instances we well, and submit positive and negative path labels to a binary classifier to can-

cel the effect of frequent but non-informative path types. Second, indiscriminate expansion around e is infeasible because the Freebase graph has very small diameter. Even after substantial pruning, paths of length 3 and 4 reach over 40% and 96% of all nodes (Fig. 3). This increases computational burden and floods us with noisy and spurious paths. We remedy the problem using an idea from PathRank (Lao and Cohen, 2010). Instead of trying to explore *all* paths originating (or terminating) at e, where e may or may not belong to a target type, we focus on paths *between* e and other known members of the target type.

3.3 Type "language model"

To exploit the second signal, shown in Fig. 1(b), we need to model the association between target YAGO types and the mention contexts of Wikipedia entities known to belong to those types. This model component is in the same spirit as (Zheng et al., 2012).

For each target YAGO type t, we sample positive entities $e \in F \cap W$, and for each e, we collect, from Wikipedia annotated text, a corpus of snippets mentioning e. We remove the mention words and retain the rest. We also collect salient words from the entire Wikipedia document containing the snippet, as shown in Fig. 1.

At this point each target type is associated with a "corpus" of contexts, each represented by snippet words. We compute the IDF of all words in this corpus[3], and then represent each type as a TFIDF vector (Salton and McGill, 1983). A test context is turned into a similar vector, and its score with respect to t is the cosine between these two vectors. This simple approach was found superior to building a more traditional smoothed multinomial unigram model (Zhai, 2008) for each type. Given the output of this component feeds into an outer discriminative inference mechanism, a strict probabilistic model is not necessary.

3.4 Entity neighborhood match with snippet

The third signal is a staple of any disambiguation work: match the occurrence context against the neighborhood in the structured representation. In word sense disambiguation (WSD), support for assigning a word in context to a synset comes from matches between, say, other words in the context and the WordNet neighborhood of the proposed synset. As in WSD, many approaches to Wikification measure some local consistency between a mention m and the neighborhood $N(e)$ of a candidate entity e. $N(e)$ is again limited by a maximum path length ℓ. From snippet m we extract all phrases $P(m)$ excluding the mention words. For each phrase $p \in P(m)$, if p occurs *at least once*[4] in any node of $N(e)$, then we accumulate a credit of $|p| \sum_{w \in p} \text{IDF}(w)$, where w ranges over words in p, $\text{IDF}(w)$ is its inverse document frequency (Salton and McGill, 1983) in Wikipedia, and $|p|$ is the length of the phrase. This rewards exact phrase matches.

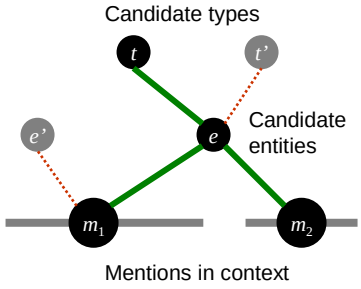

Fig. 4: Tripartite assignment problem.

4 Unified model

Figure 4 abstracts out the signals shown in Figure 1 into a tripartite assignment problem. Each mention like m_1 has to choose at most one entity from among candidate aliasing entities like e and e'. Each entity $e \in F \setminus W$ has to choose one type (for simplicity we ignore zero or more than one types as possibilities) from candidates like t and t'.

The configuration of thick (green) edges should be preferred to alternative dotted (red) edges under these considerations:

- There is high local affinity or compatibility between e and t, based on associations between t and $N(e)$ as discussed in §3.2.
- There are better textual matches between $N(e)$ and m_1, as compared to $N(e')$ and m_1.
- In aggregate, the non-mention tokens in the context of m_1, m_2 (shown as gray horizontal lines) match well the language model associated with mentions of entities of type t (rather than t').

[3] Generic IDF from Wiki text does not work.

[4] Incorporating term frequency often polluted the score.

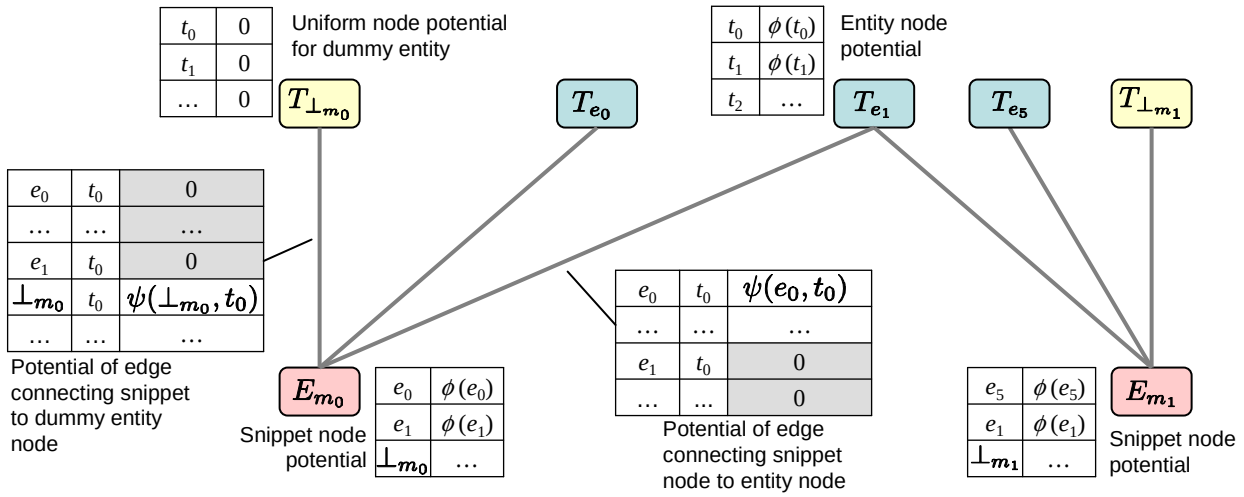

Fig. 5: Illustration of the proposed bipartite graphical model, with tables for node and edge potentials and synthetic $\perp$-entity nodes to implement the reject option.

We now present a unified model that combines the signals and solves the two proposed tasks jointly. We model two kinds of decision variables, which will be represented by nodes in a graphical model. Associated with each entity $e \in F \setminus W$ there is a hidden type variable (node) T_e, which can take on a value from (some subset of) the type catalog $\mathcal{T}$. Associated with each mention m (along with its snippet context and all its observable features) there is a hidden entity variable E_m, which can take on values from some subset of entities. (For simplicity, we assume that entities in $F \cap W$ have already been annotated in the corpus, and no m of interest mentions such entities.)

We will model the probability of a joint assignment of values to T_e, E_m as

$$\log \Pr(\vec{t}, \vec{e}) = \alpha \sum_{e} \phi_e(t_e) + \beta \sum_{m} \phi_m(e_m) + \sum_{e,m} \psi_{e,m}(t_e, e_m) - \text{const.}, \quad (1)$$

where node log potentials are called ϕ_e, ϕ_m, edge log potentials are called $\psi_{e,m}$, and α, β are tuned constants. The log partition function, written "const." above, will not be of interest in inference, where we will seek to choose $\vec{t}, \vec{e}$ to maximize $\Pr(\vec{t}, \vec{e})$. In this section, we will design the node and edge log potentials.

4.1 Node log potentials

Each node T_e is associated with a node potential table, mapping from possible types in $\mathcal{T}_e$ to nonnegative potentials. The potential values are supplied from §3.2 as the classifier output scores.

Each node E_m is associated with a node potential table, mapping from possible entities in $\mathcal{E}_m$ to nonnegative potentials. The potential values are supplied from §3.4.

4.2 Edge log potentials

Suppose we assign $E_m = e$, and $T_e = t$. Then we would like the non-mention context words of m to be highly compatible with the type "language model" developed in §3.3.

If e is among the set of values $\mathcal{E}_m$ that E_m can take, then nodes T_e and E_m are joined by an edge. This edge is associated with an edge potential table $\psi_{e,m} : \mathcal{T}_e \times \mathcal{E}_m \to \mathbb{R}^+$. $\psi_{e,m}(\cdot, e')$ will be set to zero (cells shaded gray in Fig. 5) when $e \neq e'$. $\psi_{e,m}(t, e)$ is set to the cosine match score described in §3.3.

4.3 The reject (a.k.a. null, nil, NA, $\perp$) option

An algorithm may reject many snippets, i.e., refrain from annotating them. This could be because the snippet mentions an entity outside F (and outside W), or the system wishes to ensure high precision at some cost to recall.

Rejection is modeled by adding, for each snippet m, a pseudo or "null" entity $\perp_m$ (also called "no an-

notation" NA, null or nil in the TAC-KBP[5] community). For simplicity, we assume that $\perp_m$ and $\perp_{m'}$ are incomparable or distinct for $m \neq m'$. I.e., we do not offer to cluster mentions of unknown entities. These remain separate, unconnected nodes in the (augmented) Freebase graph.

If we choose $E_m = \perp_m$, we get zero credit for matching non-mention text in m to $N(\perp_m)$, because $N(\perp_m) = \varnothing$ and §3.4 has no information to contribute. I.e., we set $\phi_m(\perp_m) = 0$. In general, $\phi_m(\cdot) \geq 0$, so $\perp_m$ gets the lowest possible credit (but this will be modified shortly).

There is also a type variable $T_{\perp_m}$. To what type should we assign "entity" $\perp_m$? Because $\perp_m$ has no connections in the Freebase graph, no hint can come from §3.2. Put differently, $\phi_{\perp_m}(t)$ will be constant (say, zero) for all snippets m and types t.

Even if we do not know the entity mentioned in m, the non-mention text in m will have differential affinity to different types, obtained from §3.3. This means that, if $E_m = \perp_m$ is chosen, $T_{\perp_m}$ will be $\arg\max_t \psi_{\perp_m,m}(t, \perp_m)$, which explains the non-mention words in m using the best available language model associated with some type. For a different entity $E_m = e_0$ and type $T_{e_0} = t$ assignment to win, $\alpha\phi_m(e_0)+\beta\phi_{e_0}(t)+\psi_{e_0,m}(t, e_0)$ must exceed the null score above. This provides a usable recall/precision handle: we modify $\phi_m(\perp_m)$ to a tuned number; making it smaller generally gives higher recall and lower precision.

4.4 Inference and training

The goal of collective inference will be to assign a type value to each T_e and an entity value to each E_m. We seek the maximum a-posteriori (MAP) label, for which we use tree-reweighted message passing (TRW-S) (Kolmogorov, 2006). Our graph has plenty of bipartite cycles, so inference is approximate. Given the sparsity of data, we preferred to *delexicalize* our objective (1), i.e., avoid word-level features and pre-aggregate their signals via time-tested aggregators (such as TFIDF cosine). As a result we have only two free parameters α, β in (1), which we tune via grid search. A more principled training regimen is left for future work.

[5] http://www.nist.gov/tac/2013/KBP/

5 Experiments

We focus our experiments on one broad type of entities, *people*, that is more challenging for disambiguators than typical showcase examples of distinguishing (Steve) Jobs from employment and Apple Inc. from fruit.

We report on three sets of experiments. In §5.1, we restrict to entities from $F \cap W$ and Wikipedia text, for which ground truth annotation is available. In §5.2, we evaluate TMI and baselines on ClueWeb12 and entities from Freebase, not limited to Wikipedia. In §5.3, we compare TMI with Google's recently published FACC1 annotations (Gabrilovich et al., 2013).

5.1 Reference corpus C_W with $F \cap W$ entities

Limited to people, $|F| = 2323792$, $|W| = 807381$, $|F \setminus W| = 1544942$, and $|F \cap W| = 778850$. It is easiest to evaluate TMI and others on Wikipedia entities. They have known YAGO types. Wikipedia text has explicit (disambiguated) entity annotations. For these reasons, the few known systems for Freebase-based annotation (Zheng et al., 2012) are evaluated exclusively on $F \cap W$.

5.1.1 Seeding and setup

People in $F \cap W$ are known by one or more mention words/phrases. From these, we collect mention phrases along with the candidate entity set for each phrase. The number of candidates is the phrase's *degree of ambiguity*. We sort phrases by ambiguity and draw a sample over the ambiguity range. This gives us seed phrases with representative ambiguity. Then we collect all entities mentioned by these phrases. Overall we collect about 1100 entities and 5500 distinct mentions.

Contrast this with (Zheng et al., 2012), who sample entities from much fewer than 130, and largely well-separated types: professional athletes, academics, actors, films, books, hotels, and tourist attractions. If there were only two namesakes, an actor and a politician, the politician disappears, leaving a naturally unambiguous alias. I.e., (Zheng et al., 2012) did not "complete" their entity sets with aliased entities. For all these reasons, Z0 numbers here are not directly comparable to those in their paper.

5.1.2 Tasks and baselines

The structure of TMI suggests two natural baselines to compare against it. TMI solves two tasks simultaneously: assign types to entities and entities to snippets. So the first baseline, **T0**, is one that solves the typing task separately, and the second, **A0**, does snippet annotation separately. A third baseline, **Z0**, from (Zheng et al., 2012) does only snippet annotation; they do not consider typing entities.

While evaluating types output by TMI and T0 against ground truth, we may wish to assign partial credit for overlapping types, e.g., *athlete* vs. *soccer player*, because our types form an incomplete hierarchy. We use the standard "M&W" score of semantic similarity between types (Milne and Witten, 2008) for this.

As regards snippet annotation, Z0 (Zheng et al., 2012) does not specify any mechanism for handling $\perp$. Therefore we run two sets of experiments. In one we eliminate all snippets with ground truth $\perp$. A0, and TMI are also debarred from returning $\perp$ for any snippet. In the other, snippets marked $\perp$ in ground truth are included. A0 and TMI are enabled to return $\perp$, but Z0 cannot.

	TMI	A0	Z0
0/1 snippet accuracy	**0.827**	0.699	0.627

Fig. 6: Snippet annotation on C_W corpus, $F \cap W$ entities, $\perp$ not allowed.

	TMI	A0	Z0
0/1 snippet accuracy	**0.7307**	0.651	0.622
Snippet precision	**0.858**	0.843	0.622
Snippet recall	**0.777**	0.692	0.639
Snippet F1	**0.815**	0.760	0.630

Fig. 7: Snippet annotation on C_W corpus, $F \cap W$ entities, $\perp$ allowed.

5.1.3 Snippet annotation results

Fig. 6 shows snippet annotation accuracy (fraction of snippets labeled with the correct entity) when $\perp$ is not allowed as an entity. As two uninformed refernces, uniform random choice gives an accuracy of 0.423 and choosing the entity with the largest prior gives an accuracy of 0.767. TMI is considerably better than A0, which is better than Z0 and the uninformed references. This is despite training Z0's per-type topic models not only on unambiguous snippets, but also on a disjoint fraction of $F \cap W$, as a surrogate for Wikipedia's containment in Freebase.

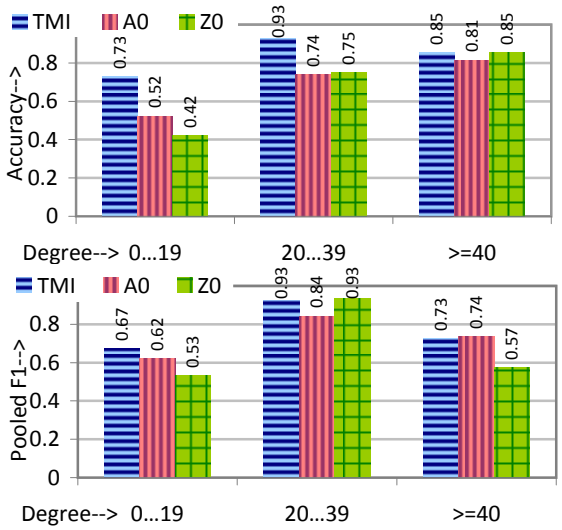

Fig. 8: Bucketed comparison between TMI and baselines, $F \cap W$, $\perp$ allowed.

	TMI	T0
0/1 type accuracy	0.80	0.81
M&W type accuracy	0.82	0.83

Fig. 9: Type inference, C_W corpus, $F \cap W$ entities.

Fig. 7 repeats the experiment while allowing $\perp$. Here, in 0/1 accuracy, $\perp$ is regarded as just another entity. Again, we see that TMI has a clear advantage. Z0's performance here is worse than in (Zheng et al., 2012). This is explained by our much larger and difficult-to-separate type system.

We disaggregate the summary results into buckets, shown in Fig. 8. Each bucket covers a range of degrees of entity nodes in Freebase, while roughly balacing the number of snippets in each bucket. TMI generally shows larger gains for low-degree buckets.

5.1.4 Type prediction results

We also compared the type inference accuracy of TMI and T0; (Zheng et al., 2012) do not infer types. The summary is in Fig. 9. Two uninformed baselines are worth mentioning. Uniform random choice over 130 types gave only 2% accuracy. Chossing the type with largest prior probability gave 28.2% accuracy. TMI is much better, but offers no significant benefit (or degradation) compared to T0. We verified, partly by way of debugging, that there do exist entities e with small degree but a modest number of assigned snippets, for which snippet-to-$N(e)$ matches

angus mcdonald, chris robinson, christopher henry, elizabeth cameron, george woods , henry barnes, jack scott, jeremy robert, john sherman, leonard thomas, marc anthony, mitchell donald, morrison mark, parker edward, richard andrew , simon scott, stephen ross, stuart baron, tom clark, whitney john, austin scott, barbara johnson, brian peterson, carlos rivero, david berman, david johns, donald fraser, george davies, george fisher, graham smith, john pepper, jonathan edwards, kevin brown, kevin hughes, matt johnson, michael davidson, nancy johnson, paul holmes, pedro martins, peter frank, peter mitchell, peter mullen, robert stern, roger edwards, stuart walker, terry evans, tony angelo, tony ward, william jarvis, william sampson

Fig. 10: Seed mentions for confusion clusters for Web corpus C and entities in F.

provide a boost to type prediction accuracy (about 50%), as compared to T0 (about 20% for these instances). Therefore, the flow of information between type and entity assignments is, in principle, bidirectional, in the regime of such entities.

5.2 Payload corpus C with entities in F

Recall our main goal is to annotate payload corpus C with entities in all of F. Experience with $F \cap W$ and C_W may not be representative of F and C. Entities in $F \setminus W$ may not come with reference mentions, and their type and entity neighborhoods may be sparse. Furthermore, compared to the closed world of $F \cap W$, evaluating TMI and baselines over C and $F \setminus W$ is challenging. Entities in $F \setminus W$ do not have ground truth types in the type catalog T (here, YAGO), nor snippets labeled by humans as mentioning them. Therefore, we need human editorial judgment, which is scarce. Even though TMI can be *applied* at Web scale, the scale if *evaluation* is limited by editorial input.

5.2.1 Seeding and data setup

There are about 2.3 million Freebase entities connected to */people/person* via *type* links. Similar to §5.1.1, we chose phrases (Fig. 10) with diverse degree of ambiguity (Fig. 11), to seed confusion clusters. Then we completed the clusters by including aliased entities, as before, so as not to artifically reduce the degree of ambiguity. Note that entities in W can and do contend with entities in $F \setminus W$. The cluster size distribution is shown in Fig. 12. Limited by editorial budget, we finished with 634 entities, 238 distinct aliases and 4,500 snippets.

We used the 700-million-page ClueWeb12 Web corpus. All phrases in the expanded clusters are

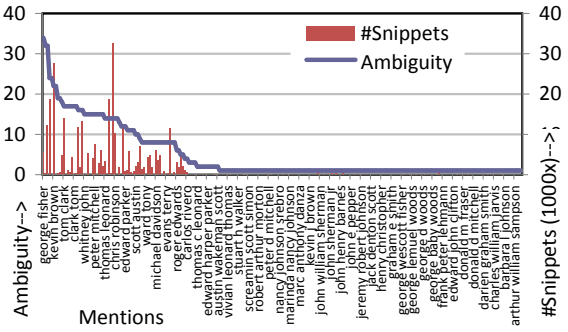

Fig. 11: Ambiguity distribution for Web corpus C. Unambiguous names are usually fully expanded and very rare, if at all present, in evaluated snippets.

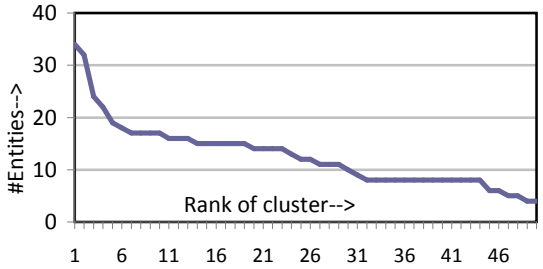

Fig. 12: Confusion cluster size distribution for Web corpus C.

loaded into a trie used by a map-reduce job to extract documents, then snippets, from the corpus. Some phrases in Figs. 10 and 11 have overwhelming numbers of pages with matches. In production, we naturally want all of them to be annotated. But human editorial judgment being the bottleneck, we sampled 50% or 50,000 snippets, whichever was smaller. Starting with about 752,450 pages, we ran the Stanford NER (Finkel et al., 2005) to mark person spans. Pages with fewer than five non-person tokens per person were discarded; this effectively discarded long list pages without any informative text to disambiguate anyone, and left us with 574,135 pages. From these we collected 304,309 snippets where the mention phrase is marked by the NER as a person. Each seed phrase leads to one cluster on which TMI and A0 are run. Note that $\perp$ must be allowed on the open Web.

5.2.2 Editorial judgment

Finally, for each algorithm, about 634 entity-type and about 4500 snippet-entity assignments are randomly sampled and sent to 20 editors in a commercial search engine company, who judged each assignment as correct or incorrect, *without knowing which algorithm* produced the annotation, to avoid

	TMI entity	A0
0/1 accuracy	**.714**	.562
Pooled recall	.764	**.869**
Precision	**.714**	.562
F1	**.738**	.683

	$e \in W$ (0/1 acc.)	$e \in F \setminus W$ (0/1)
TMI	**.75**	**.62**
A0	.65	.42

Fig. 13: Snippet summary for F and payload corpus C.

bias. Because the editors are trained professionals (unlike Mechanical Turks), we increased our evaluation coverage by having each type or entity assignment reviewed by one editor.

Pooling: Ideally, editors can be asked to find the best type or entity for each entity or snippet, but, given the size and diversity of Freebase, the cognitive burden would be unacceptable. In the Wikipedia corpus C_W, a snippet marked $\perp$ (no entity) by an algorithm can be judged a loss of recall if Wikipedia ground truth annotates it with an entity. Unfortunately, this is no longer practical for Web corpus C, because 8,217 snippets marked $\perp$ would have to be manually inspected and compared with a large number of candidate entities in Freebase. Therefore, we adopt pooling as in TREC. (Although the pool is small, A0 has very high recall.) Recall is evaluated with respect to the union of snippets annoted with a non-$\perp$ entity by *at least one* competing algorithm, with agreement in case of more than one.

0/1 Type accuracy: Editors judged each proposed type as correct or not. Unlike in §5.1, where the true and proposed types could be compared via M&W (Milne and Witten, 2008), they could not be asked to objectively estimate relatedness between types. Therefore we present only their reported post-hoc 0/1 accuracy for types: T0 and TMI have 0/1 type accuracy of 0.828 and 0.818.

5.2.3 Snippet annotation results

Given the large gap between TMI and Z0 in the easier setup in §5.1, we no longer consider Z0, and instead focus on TMI vs. A0. The summary comparison of A0 vs. TMI is shown in Fig. 13. Here TMI's absolute gains in 0/1 accuracy and F1 are even larger than in §5.1. To understand TMI's performance across a diversity of Freebase entity nodes e, as a function of 1. the size and richness of $N(e)$, and 2. the number of snippets claimed to mention e, we disaggregate the data of Fig. 13 into buckets of

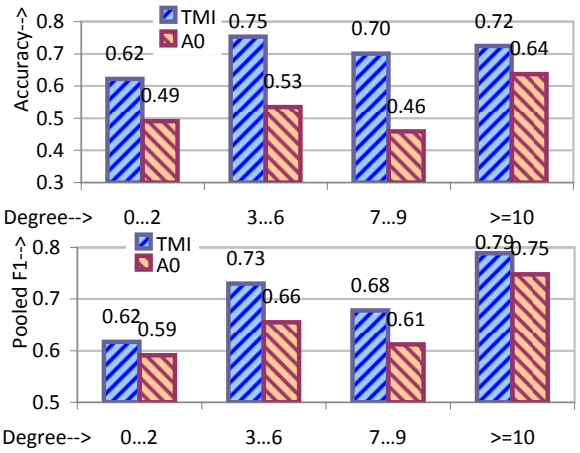

Fig. 14: 0/1 accuracy and F1 for snippets, payload corpus C and entities in F.

Snippet label judgements	%
TMI ok, FACC1 ok, neither $\perp$	22
TMI ok, FACC1 wrong, neither $\perp$	6
TMI$\neq \perp$ ok, FACC1=$\perp$ wrong	**40**
TMI$\neq \perp$ wrong, FACC1=$\perp$	23
TMI wrong, FACC1 wrong, neither $\perp$	2
TMI$= \perp$ wrong, FACC1$\neq \perp$ correct	4
TMI$= \perp$, FACC1$\neq \perp$, wrong	3
(TMI$= \perp$, FACC1$= \perp$, not judged)	-

Fig. 15: TMI vs. FACC1 comparison.

consecutive degrees, roughly balancing the number of snippets per bucket, as shown in Fig. 14. At the very low end of almost disconnected entity nodes, no algorithm does very well, because these entities are also hardly ever mentioned. When the entity is popular and well-connected, TMI's benefits are relatively modest. TMI's gains are best in the mid-range of degrees. The gap narrows for large-degree nodes, which is expected.

5.3 Comparison with FACC1

After collecting our pool of snippets as in §5.2.2, we consulted FACC1 (Gabrilovich et al., 2013), and passed on FACC1 annotations to our editors. As before, the identity of the algorithm was concealed. Results are shown in Fig. 15. In a large 40% of cases, TMI labels correctly while FACC1 backs off. The converse, where FACC1 backs off and TMI makes a mistake, is about half as frequent. These preliminary numbers suggest that TMI is able to push recall beyond FACC1 while also giving better precision.

6 Conclusion

We presented a formal model for bootstrapping from YAGO types and entities annotated in Wikipedia to two tasks, 1. annotating Web snippets with Freebase entities, and 2. associating Freebase entities with YAGO types. We presented TMI, a system to solve the two tasks jointly. Experiments show that TMI's snippet annotation accuracy, especially for relatively weakly-connected Freebase entities, is superior to baselines. We aim to extend from people to all major Freebase categories, and larger Web crawls.

Acknowledgment: We are grateful to Shrikant Naidu, Muthusamy Chelliah, and the editors from Yahoo! for their generous support. Shashank Gupta helped process FACC1 data.

References

S. Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *EMNLP Conference*, pages 708–716.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL Conference*, pages 363–370.

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. FACC1: Freebase annotation of ClueWeb corpora. `http://lemurproject.org/clueweb12/`, June. Version 1 (Release date 2013-06-26, Format version 1, Correction level 0).

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *SIGIR Conference*, pages 267–274. ACM.

Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in Web text: A graph-based method. In *SIGIR Conference*, pages 765–774.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *EMNLP Conference*, pages 782–792, Edinburgh, Scotland, UK, July. SIGDAT.

Vladimir Kolmogorov. 2006. Convergent tree-reweighted message passing for energy minimization. *IEEE PAMI*, 28(10):1568–1583, October.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of Wikipedia entities in Web text. In *SIGKDD Conference*, pages 457–466.

Ni Lao and William W. Cohen. 2010. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67, October.

Xiaonan Li, Chengkai Li, and Cong Yu. 2010. EntityEngine: Answering entity-relationship queries using shallow semantics. In *CIKM*, October. (demo).

Thomas Lin, Mausam, and Oren Etzioni. 2012. No noun phrase left behind: detecting and typing unlinkable entities. In *EMNLP Conference*, pages 893–903.

R Mihalcea and A Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *CIKM*, pages 233–242.

David Milne and Ian H Witten. 2008. Learning to link with Wikipedia. In *CIKM*, pages 509–518.

Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. 2013. Fine-grained semantic typing of emerging entities. In *ACL Conference*.

Patrick Pantel, Thomas Lin, and Michael Gamon. 2012. Mining entity types from query logs via user intent modeling. In *ACL Conference*, pages 563–571, Jeju Island, Korea, July.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. In *ACL Conference*, ACL/HLT, pages 1375–1384, Portland, Oregon.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: an experimental study. In *EMNLP Conference*, pages 1524–1534, Edinburgh, UK. ACL.

G Salton and M J McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill.

Uma Sawant and Soumen Chakrabarti. 2013. Learning joint query interpretation and response ranking. In *WWW Conference*, Brazil.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A core of semantic knowledge unifying WordNet and Wikipedia. In *WWW Conference*, pages 697–706. ACM Press.

ChengXiang Zhai. 2008. Statistical language models for information retrieval: A critical review. *Foundations and Trends in Information Retrieval*, 2(3):137–213, March.

Zhicheng Zheng, Xiance Si, Fangtao Li, Edward Y. Chang, and Xiaoyan Zhu. 2012. Entity disambiguation with Freebase. In *Web Intelligence Conference*, WI-IAT '12, pages 82–89.