No Noun Phrase Left Behind: Detecting and Typing Unlinkable Entities

Thomas Lin, Mausam, Oren Etzioni
Computer Science & Engineering
University of Washington
Seattle, WA 98195, USA
{tlin, mausam, etzioni}@cs.washington.edu

Abstract

Entity linking systems link noun-phrase mentions in text to their corresponding Wikipedia articles. However, NLP applications would gain from the ability to detect and type all entities mentioned in text, including the long tail of entities not prominent enough to have their own Wikipedia articles. In this paper we show that once the Wikipedia entities mentioned in a corpus of textual assertions are linked, this can further enable the detection and fine-grained typing of the unlinkable entities. Our proposed method for detecting unlinkable entities achieves 24% greater accuracy than a Named Entity Recognition baseline, and our method for fine-grained typing is able to propagate over 1,000 types from linked Wikipedia entities to unlinkable entities. Detection and typing of unlinkable entities can increase yield for NLP applications such as typed question answering.

1 Introduction

A key challenge in machine reading (Etzioni et al., 2006) is to identify the entities mentioned in text, and associate them with appropriate background information such as their type. Consider the sentence "Some people think that pineapple juice is good for vitamin C." To analyze this sentence, a machine should know that "pineapple juice" refers to a beverage, while "vitamin C" refers to a nutrient.

Entity linking (Bunescu and Paşca, 2006; Cucerzan, 2007) addresses this problem by linking noun phrases within the sentence to entries in a large, fixed entity catalog (almost always Wikipedia). Thus, entity linking has a limited and somewhat arbitrary range. In our example, systems by (Ferragina and Scaiella, 2010) and (Ratinov et al., 2011) both link "vitamin C" correctly, but link "pineapple juice" to "pineapple." "Pineapple juice" is not entity linked as a beverage because it is not prominent enough to have its own Wikipedia entry. As Table 1 shows, Wikipedia often has prominent entities, while missing tail and new entities of the same types.[1] (Wang et al., 2012) notes that there are more than 900 different active shoe brands, but only 82 exist in Wikipedia. In scenarios such as intelligence analysis and local search, non-Wikipedia entities are often the most important.

example noun phrases	status
"apple juice" "orange juice"	present
"prune juice" "wheatgrass juice"	**absent**
"radiation exposure" "workplace stress"	present
"asbestos exposure" "financial stress"	**absent**
"IJCAI" "OOPSLA"	present
"EMNLP" "ICAPS"	**absent**

Table 1: Wikipedia has entries for prominent entities, while missing tail and new entities of the same types.

Hence, we introduce the *unlinkable noun phrase problem*: Given a noun phrase that does not link into Wikipedia, return whether it is an entity, as well its fine-grained semantic types. Deciding if a non-Wikipedia noun phrase is an entity is challenging because many of them are not entities (e.g., "Some people," "an addition" and "nearly half"). Predict-

[1] The same problem occurs with Freebase, which is also missing the same Table 1 entities.

ing semantic types is a challenge because of the diversity of entity types in general text. In our experiments, we utilized the Freebase type system, which contains over 1,000 semantic types.

The first part of this paper proposes a novel method for detecting entities by observing that entities often have different usage-over-time characteristics than non-entities. Evaluation shows that our method achieves 24% relative accuracy gain over a NER baseline. The second part of this paper shows how instance-to-instance class propagation (Kozareva et al., 2011) can be adapted and scaled to semantically type general noun-phrase entities using types from linked entities, by leveraging over one million different possible textual relations.

Contributions of our research include:

- We motivate and introduce the *unlinkable noun phrase problem*, which extends previous work in entity linking.
- We propose a novel method for discriminating entities from arbitrary noun phrases, utilizing features derived from Google Books ngrams.
- We adapt and scale instance-to-instance class propagation in order to associate types with non-Wikipedia entities.
- We implement and evaluate our methods, empirically verifying improvement over appropriate baselines.

2 Background

In this section we provide an overview of entity linking, how we entity link our data set, and describe how our problem and approach differ from related areas such as NER and Web extraction.

2.1 Entity Linking

Given text, the task of entity linking (Bunescu and Paşca, 2006; Cucerzan, 2007; Milne and Witten, 2008; Kulkarni et al., 2009) is to identify the Wikipedia entities within the text, and mark them with which Wikipedia entity they correspond to. Entity linking elevates us from plain text into meaningful *entities* that have properties, semantic types, and relationships with each other. Other entity catalogs can be used in place of Wikipedia, especially in domain-specific contexts, but general purpose linking systems all use Wikipedia because of its broad general coverage, and to leverage its article texts and link structure during the linking process.

A problem we observed when using entity linking systems is that despite containing over 3 million entities, Wikipedia does not cover a significant number of entities. This occurs with entities that are not prominent enough to have their own dedicated article and with entities that are very new. For example, Facebook has over 600 million users, and each of them could be considered an entity. The REVERB extractor (Fader et al., 2011) on the ClueWeb09 Web corpus found over 1.4 billion noun phrases participating in textual relationships, and a sizable portion of these noun phrases are entities. While recent research has used NIL features to determine whether they are being asked to link an entity not in Wikipedia (Dredze et al., 2010; Ploch, 2011), there has been no research on whether given noun phrases that are *unlinkable* (for not being in Wikipedia) are entities, and how to make them usable if they are.

Our goal is to address this problem of learning whether non-Wikipedia noun phrases are entities, and assigning semantic types to them to make them useful. We begin with a corpus of 15 million "(noun phrase subject, textual relation, noun phrase object)" assertions from the Web that were extracted by REVERB (Fader et al., 2011).[2] REVERB already filters out relative pronouns, WHO-adverbs, and existential "there" noun phrases that do not make meaningful arguments. We then employ standard entity linking techniques including string matching, prominence priors (Fader et al., 2009), and context matching (Bunescu and Paşca, 2006) to link the noun phrase subjects into Wikipedia.

In this manner, we were able to entity link the noun phrase subject of 9,699,967 extractions, while the remaining 5,028,301 extractions had no matches (mostly due to no close string matches). There were 1,401,713 distinct noun phrase subjects in the 5 million extractions that had no matches. These are the *unlinkable* noun phrases we will study here.

2.2 Named Entity Recognition

Named Entity Recognition (NER) is the task of identifying named entities in text. A key difference between our final goals and NER is that in the con-

[2] available at http://reverb.cs.washington.edu

text of entity linking and Wikipedia, there are many more entities than just the *named* entities. For example, "apple juice" and "television" are Wikipedia entities (with Wikipedia articles), but are not traditional named entities. Still, as named entities do comprise a sizable portion of our *unlinkable* noun phrases, we compare against a NER baseline in our entity detection step.

Fine-grained NER (Sekine and Nobata, 2004; Lee et al., 2007) has studied scaling NER to up to 200 semantic types. This differs from our semantic typing of unlinked entities because our approach assumes access to corpora-level relationships between a large set of linked entities (with semantic types) and the unlinked entities. As a result we are able to propagate 1,339 Freebase semantic types from the linked entities to the unlinked entities, which is substantially more types than fine-grained NER.

2.3 Extracting Entity Sets

There is a line of research in using Web extraction (Etzioni et al., 2005) and entity set expansion (Pantel et al., 2009) to extract lists of typed entities from the Web (e.g., a list of every city). Our problem instead focuses on determining whether any individual noun phrase is an entity, and what semantic types it holds. Given a noun phrase representing a person name, we return that this is a person entity even if it is not in a list of people names harvested from the Web.

3 Architecture

Our goal is: given (1) a large set of linked assertions L and (2) a large set of unlinked assertions U, for each unlinkable noun phrase subject $n \in U$, predict: (1) whether n is an entity, and if so, then (2) the set of Freebase semantic types for n. For L we use the 9.7 million assertions whose *subject* argument we were able to link in Section 2.1, and for U we use the 5 million assertions that we could not link.

We divide the system into two components. The first component (described in Section 4) takes any unlinkable noun phrase and outputs whether it is an entity. All $n \in U$ classified as entities are placed in a set E. The second component (described in Section 5) uses L and U to predict the semantic types for each entity $e \in E$.

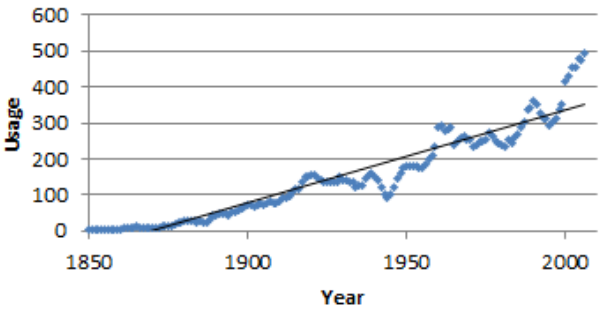

Figure 1: Usage over time in Google books for the noun phrase "Prices quoted" (e.g., from "Prices quoted are for 2 adults") which is not an entity.

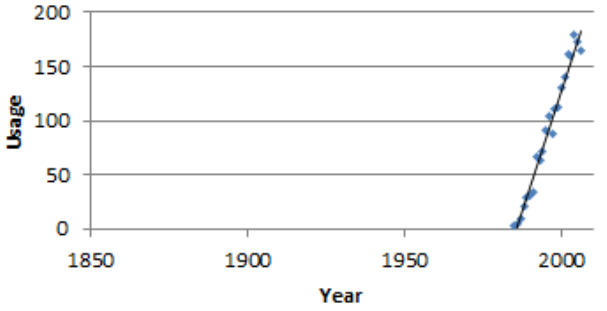

Figure 2: Usage over time for the unlinkable noun phrase "Soluble fibre," which is an entity. The best fit line has steeper slope compared to Figure 1.

4 Detecting Unlinkable Entities

This first task takes in any *unlinkable* noun phrase and outputs whether it is an entity. There is a long history of discussion in analytic philosophy literature on the question of what exists (e.g., (Quine, 1948)). We adopt a more pragmatic view, defining an "entity" as a noun phrase that could have a Wikipedia-style article if there were no notability or newness considerations, and which would have semantic types. We are interested in entities that could help populate an entity store. "EMNLP 2012" is an example of an entity, while "The method" and "12 cats" are examples of noun phrases that are not entities. This is challenging because at a surface level, many entities and non-entities look similar: "Sex and the City" is an entity, while "John and David" is not. "Eminem" is an entity, while "Youd" (a typo from "You'd") is not.

We address this task by training a classifier with features primarily derived from a timestamped corpus. An intuition here is that when considering only unlinkable noun phrases, usage patterns across

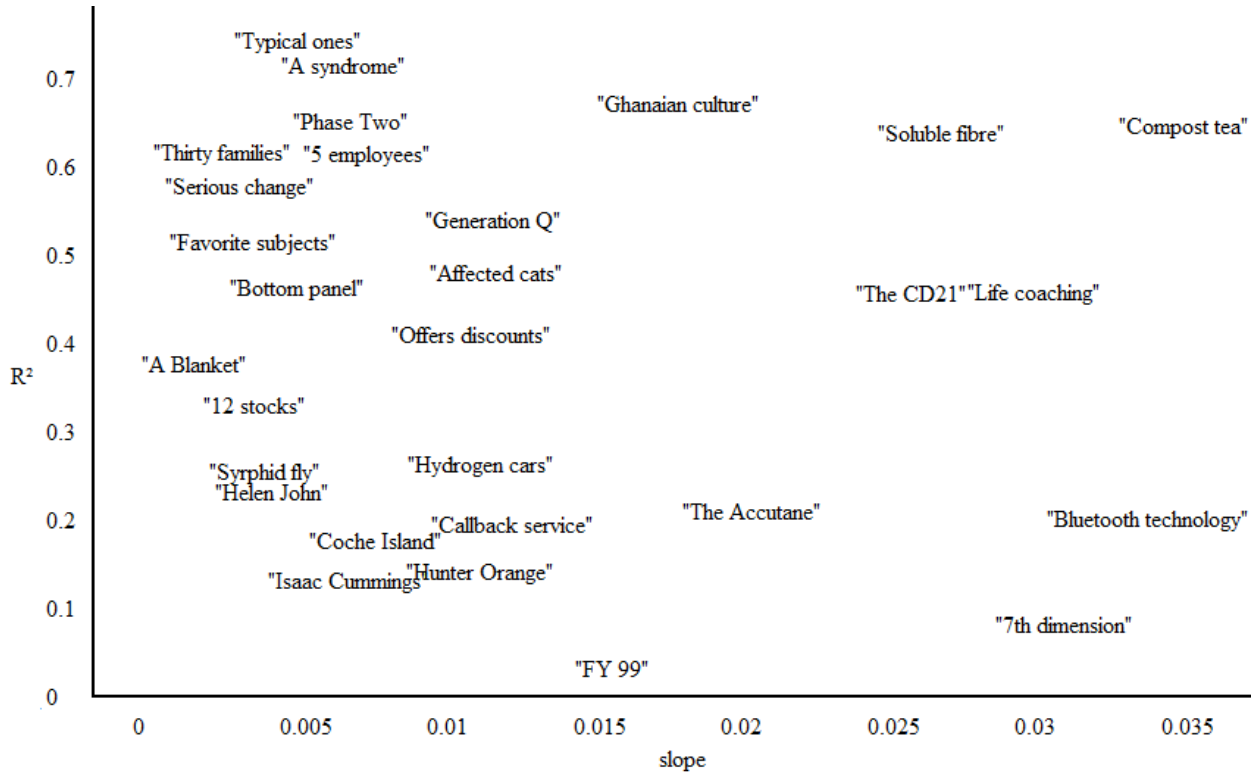

Figure 3: Plot of R^2 vs $Slope$ for the usage over time of a collection of noun phrases selected for illustrative purposes. Many of the non-entities occur at lower $Slope$ and higher R^2, while the entities often have higher slope and/or lower R^2. "Bluetooth technology" actually has even higher slope, but was adjusted left to fit in this figure.

time often differ for entities and non-entities. Noun phrase entities that are observed in text going back hundreds of years (e.g., "Europe") almost all have their own Wikipedia entries, so in unlinkable noun phrase space, the remaining noun phrases that are observed in text going back hundreds of years tend to be all the textual references and expressions that are not entities. We plan to use this signal to help separate the entities from the non-entities.

4.1 Classifier Features

We use the Google Books ngram corpus (Michel et al., 2010), which contains timestamped usage of 1-grams through 5-grams in millions of digitized books for each year from 1500 to 2007.[3] We use ngram match count values from case-insensitive matching. To avoid sparsity anomalies we observed in years before 1740, we use the data from 1740 onward. While it has not been used for our task before, this corpus is a rich resource that enables reasoning about knowledge (Evans and Foster, 2011) and understanding semantic changes of words over time (Wijaya and Yeniterzi, 2011). Talukdar et al. (2012) recently used it to effectively temporally scope relational facts.

Our first feature is the *slope* of the least-squares best fit line for usage over time. For example, if a term appears 25 times in books in 1950, 30 times in 1951, ..., 100 times in 2007, then we compute the straight line that best fits {(1950, 25), (1951, 31), ..., (2007, 100)}, and examine the slope. We have observed cases of non-entity noun phrases (e.g., Figure 1) having lower slopes than entity noun phrases (e.g., Figure 2). Note that we do not normalize match counts by yearly total frequency, but we do normalize counts for each term to range from 0 to 1 (setting the max count for each term to 1) to avoid bias from entity prominence. To capture the current usage, in cases where there exists a ≥ 5 year gap in usage of a term we only use the data after the gap.

Another feature is the R^2 fit of the best fit line. Higher R^2 indicates that the data is closer to a straight line. Figure 3 plots R^2 vs $Slope$ values

[3] available at http://books.google.com/ngrams/datasets

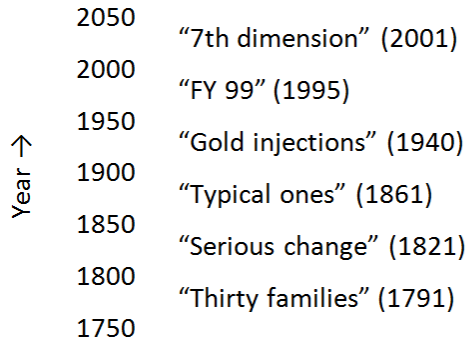

Figure 4: *UsageSinceYear* of example unlinked terms.

for some sample noun phrases. We observed that along with their lower $Slope$, the non-entities often also had higher R^2, indicating that their usage does not vary wildly from year to year. This contrasts with certain entities (e.g., "FY 99" for "Fiscal Year 1999") whose usage sometimes varied sharply from year to year based on their prominence in those specific years.

A third feature is *UsageSinceYear*, which finds the year from when a term last started continually being used. For example, a *UsageSinceYear* value of 1920 would indicate that the term was used in books every year from 1920 through 2007. Figure 4 shows examples of where various terms fall along this dimension.

From the books ngram corpus, we also calculate features for: *PercentProperCaps* - the percentage of case-insensitive matches for the term where all words began with a capital letter, *PercentExactMatch* - the percentage of case-insensitive matches for the term that matched the capitalization in the assertion exactly, and *Frequency* - the total number of case-insensitive occurrences of the term in the book ngrams data, summed across all years, which reflects prominence. Last, we also include a simple *numeric* feature to detect the presence of leading numeric words (e.g., "5" in "5 days" or "Three" in "Three choices").

4.2 Evaluation

From the corpus of 15 million REVERB assertions, there were 1.4 million unlinked noun phrases including 17% unigrams, 51% bigrams, 21% trigrams, and 11% 4-grams or longer. Bigrams comprise over half the noun phrases and the books bigram data is a self-contained download that is easier to obtain and store than the full books ngram corpus, so we focus on bigrams in our evaluation. In a random sample of unlinked bigrams, we found that 73% were present in the books ngram data (65% exact match, 8% case-insensitive match only), while 27% were not (these were mostly entities or errors with non-alphabetic characters). Coverage is a greater issue with longer ngrams (e.g., there are many more possible 5-grams than bigrams, so any individual 5-gram is less likely to reach the minimum threshold to be included in the books data), but as mentioned earlier, only 11% of unlinkable noun phrases were 4-grams or longer.

system	correctly classified
Majority class baseline	50.4%
Named Entity Recognition	63.3%
Slope feature only	61.1%
PUF feature combination	69.1%
ALL features	78.4%

Table 2: Our classifier using all features (*ALL*) outperforms majority class and NER baselines.

We randomly sampled 250 unlinked bigrams that had books ngram data, and asked 2 annotators to label each as "entity," "non-entity," or "unclear." Our goal is to separate noun phrases that are clearly entities (e.g., "prune juice") from those that are clearly not entities (e.g., "prices quoted"), rather than to debate phrases that may be in some entity store definitions but not others, so we asked the annotators to choose "unclear" when there was any doubt. There were 151 bigrams that both annotators believed to be very clear labels, including 69 that both annotators labeled as entities, 70 that both annotators labeled as non-entities, and 12 with label disagreement. Cohen's kappa was 0.84, indicating excellent agreement. Our experiment is now to separate the 69 clear entities from the 70 clear non-entities.

For experiment baselines we use the majority class baseline *MAJ*, as well as a Named Entity Recognition baseline *NER*. For *NER* we used the Illinois Named Entity Tagger (Ratinov and Roth, 2009) on the highest setting (that achieved 90.5 F_1 score on the CoNLL03 shared task). *NER* expects a sentence, so we use the longest assertion in the corpus that the noun phrase was observed in. We evaluate several combinations of our features to test dif-

ferent aspects of our system: *Slope* uses only *Slope*, *PUF* uses *PercentProperCaps* + *UsageSinceYear* + *Frequency*, and *ALL* uses all features. We evaluate using the WEKA J48 Decision Tree on default settings, with leave-one-out cross validation.

Table 2 shows the results. *MAJ* correctly classifies 50.4% of instances, *NER* correctly classifies 63.3% and *ALL* correctly classifies 78.4%.

4.3 Analysis

Overall, 78.4% correctly classified instances is fairly strong performance on this task. By using the described features, our classifier was able to detect and filter many of the non-entity noun phrases in this scenario. Compared to the 63.3% of *NER*, it is an absolute gain of 15.1%, a relative gain of 24%, and a reduction in error of 41.1% (from 36.7% to 21.6%). Student's t-test at 95% confidence verified that the difference was significant.

We found that while low *Slope* (especially with higher R^2) often indicated non-entity, there were numerous cases where higher *Slope* did not necessarily indicate entity. For example, the noun phrase "several websites" has fairly sharp slope, but still does not denote a clear entity. In these cases, the addition of other features can serve as additional useful signal. One error from *ALL* is the term "Analyst estimates," which the annotators labeled as a non-entity, but which occasionally appears in text (especially titles) as "Analyst Estimates," and is a relatively recent phrase. *NER* misses entities such as "synthetic cubism" and "hunter orange" that occur in our data but are not traditional named entities. We observed that while none of our features achieves over 70% accuracy by themselves, they perform well in conjunction with each other.

5 Propagating Semantic Types

This second task uses a set of linked assertions *L* and set of unlinked assertions *U* to predict the semantic types for each entity $e \in E$. If the previous step output that "Sun Microsystems" is likely to be an entity, then the goal of this step is to further predict that it has the Freebase types such as *organization* and *software developer*.

From *L* we use the set of linked entities and the textual relations they occur with. For example, *L* might contain that the entity *Microsoft* links to a particular Wikipedia article, and also that it occurs with textual relations such as "has already announced" and "has released updates for." For each Wikipedia-linked entity in L, we further look up its exact set of Freebase types.[4] From U we obtain the set of textual relations that each $e \in E$ is in the domain of. We now have a large set of class-labeled instances (all entities in *L*), a large set of unlabeled instances (E), and a method to connect the unlabeled instances with the class-labeled instances (via any shared textual relations), so we cast this task as an instance-to-instance class propagation problem (Kozareva et al., 2011) for propagating class labels from labeled to unlabeled instances.

We build on the recent work of Kozareva et al. (2011), and adapt their approach to leverage the scale and resources of our scenario. While they use only one type of edge between instances, namely shared presence in the high precision *DAP* pattern (Hovy et al., 2009), our final system uses 1.3 million textual relations from $|L \cup U|$, corresponding to 1.3 million potential edge types. Their evaluation involved only 20 semantic classes, while we use all 1,339 Freebase types covered by our entities in L.

There is a rich history of other approaches for predicting semantic types. (Talukdar et al., 2008) and (Talukdar and Pereira, 2010) model relationships between instances and classes, but our unlinked entities do not come with any class information. Pattern-based approaches (Paşca, 2004; Pantel and Ravichandran, 2004) are popular, but (Kozareva et al., 2011) notes that they "are constraint to the information matched by the pattern and often suffer from recall," meaning that they do not cover many instances. Classifiers have also been trained for fine-grained semantic typing, but for noticeably fewer types than we work with. (Rahman and Ng, 2010) studied hierarchical and collective classification using 92 types, and FIGER (Ling and Weld, 2012) recently used an adapted perceptron for multi-class multi-label classification into 112 types.

5.1 Algorithm

Given an entity *e*, our algorithm involves: (1) finding the textual relations that *e* is in the domain of, (2)

[4] data available at http://download.freebase.com/wex

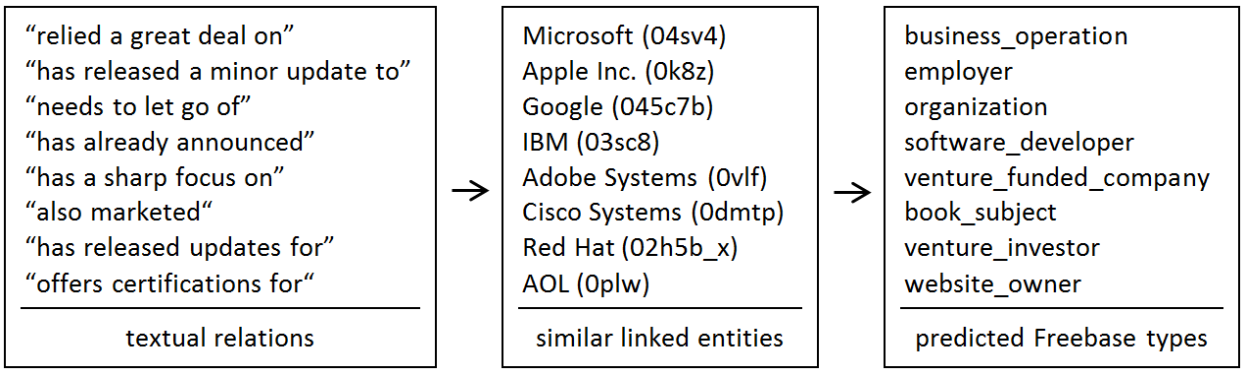

Figure 5: This example illustrates the set of Freebase type predictions for the noun phrase "Sun Microsystems." We predict the semantic type of a noun phrase by: (1) finding the textual relations it is in the domain of, (2) finding linked entities that are also in the domain of those textual relations, and (3) observing their semantic types.

finding linked entities that are also in the domain of those textual relations, and then (3) predicting types by observing the types of those linked entities. Figure 5 illustrates how we would predict the semantic types of the noun phrase "Sun Microsystems."

Find Relations: Obtain the set R of all textual relations in U that e is in the domain of. For example, if U contains the assertion "(Sun Microsystems, has released a minor update to, Java 1.4.2)," then the textual relation "has released a minor update to" should be added to R when typing "Sun Microsystems."

Find Similar Entities: Find the linked entities in L that are in the domain of the most relations in R. In our example, entities such as "Microsoft" and "Apple Inc." have the greatest overlap in textual relations because they are most often in the domain of the same textual relations, e.g., ("Microsoft, has released a minor update to, Windows Live Essentials"). Create a set S of the entities that share the most textual relations. We found keeping 10 similar entities ($|S|$ = 10) is generally enough to predict the original entity's types in the final step.

Predict Types: Return the most frequent Freebase types of the entities in S as the prediction. To avoid penalizing very small types, if there are n instances of semantic class C in S, then we rank C using a type score $T(n, C, S) = max(n/|S|, n/|C|)$, which we found to perform better than $T(n, C, S) = avg(n/|S|, n/|C|)$. For "Sun Microsystems," *business operation* was the top predicted type because all entities in S were observed to include *business operation* type.

5.2 Edge Validity

This algorithm will only be effective if entities that share textual relation strings are more likely to be of the same semantic types. To verify this, we sampled 30,000 linked entities from L that had at least 30 textual relations each, and associated each with their 30 most frequent relations. From the 900 million possible entity pairs, we then randomly sample 500 entity pairs that shared exactly *k* out of 30 relations, for each *k* from 0 to 15. At each *k* we then use our sampled pairs to estimate the probability that any two entities sharing exactly *k* relations (out of their 30 possible) will share at least one type.

The results are shown in Figure 6. We found that entities sharing more textual relations were in fact more likely to have semantic types in common. Two entities that shared exactly 0 of 30 textual relations were only 11% likely to share a semantic type, while two entities that shared exactly 10 of 30 relations were 80% likely to share a semantic type. This validates our use of textual relations as a signal-bearing edge in instance-to-instance class propagation.

5.3 Weighting Textual Relations

The algorithm as currently described treats all textual relations equally, when in reality some are stronger signal to entity type than others. For example, two entities in the domain of the "came with" relation often will not share semantic types, but two entities in the domain of the "autographed" relation will almost always share a type. To capture this intuition, we define relation weight $w(r)$ as the observed probability (among the linked entities) that two en-

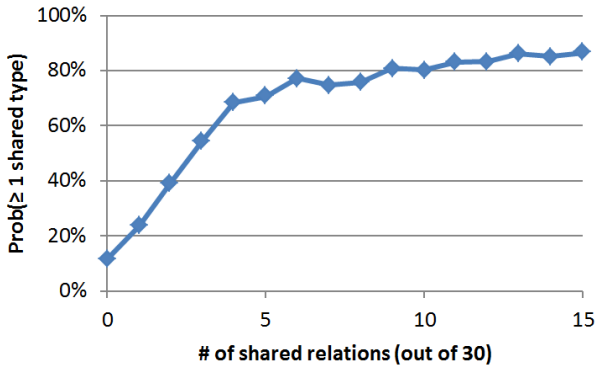

Figure 6: Entities that share more textual relations are more likely to have semantic types in common.

"is a highway in"
"is a university located in"
"became the president of"
"turned down the role of"
"has an embassy in"

Table 3: Example relations found to have high weight.

"comes with"
"is a generic term for"
"works best on"
"can be made from"
"is almost identical to"

Table 4: Example relations found to have low weight.

tities will share a Freebase type if they both occur in the domain of r. If $D(r)$ = all entities observed in the domain of relation r and $T(e)$ = all Freebase types listed for entity e, then weight $w(r)$ of a textual relation string r is:

$$w(r) = \sum_{e_1,e_2 \in D(r),\ e_1 \neq e_2} \frac{I(e_1,e_2)}{|D(r)| \cdot (|D(r)|-1)}$$

$$I(e_1,e_2) = \begin{cases} 1, & \text{if } |T(e_1) \cap T(e_2)| > 0 \\ 0, & \text{otherwise.} \end{cases}$$

Table 3 shows examples of high weight relations, and Table 4 shows low weight relations. We now modify the *Find Similar Entities* step such that if a linked entity shares a set of relations Q with the entity being typed, then it receives a score which considers all shared relations $q \in Q$ but uses the high weight relations more. On a development set we found that a score of $\sum_{q \in Q} 10^{4 \cdot w(q)}$ was effective, as higher weight signifies much stronger signal. This score then determines which entities to place in S.

5.4 Evaluation

The goal of the evaluation is to judge how well our method can predict the Freebase semantic types of entities in our scenario. Our linked entities covered 1,339 Freebase types, including many interesting types such as *computer operating system*, *religious text*, *airline* and *baseball team*. Human judges would have trouble manually annotating new entities with all these types because there are too many to keep in mind and understand the characteristics of. Instead, we automatically generate testing data by sampling entities from L, and then test on ability to recover the actual Freebase types (which we know).

We sample a *HEAD* set of distinct 500 Freebase entities (drawn randomly from our set of linked extractions), and a *TAIL* set of 500 entities (drawn randomly from our set of linked entities). An entity that occurs in many extractions is more likely to be in *HEAD* than *TAIL*. Our sampling also picks only entities that occur with at least 10 relations, which is appropriate for the Web scenario where more instances can always be queried for.

For baselines we use random baseline B_{Random} and a frequency baseline $B_{Frequency}$ which always returns types in order of their frequency among all linked entities (e.g., always *person*, then *location*, etc). We evaluate our system without relation weighting ($S_{NoWeight}$) and also with relation weighting ($S_{Weighted}$). For $S_{Weighted}$ we leave all the test set entities out when calculating global relation weights. Our metrics are *Precision at 1* and F_1 *score*. Precision at 1 measures how often the top returned type is a correct type, and is useful for applications that want one type per entity. F_1 measures how well the method recovers the full set of Freebase types (for each test case we graph precision/recall and take the max F_1), and is useful for applications such as typed question answering.

Table 5 shows the results. B_{Random} performs poorly because there are so many semantic types, and very few of them are correct for each test case. $B_{Frequency}$ performs slightly better on *TAIL* than *HEAD* because *TAIL* contains more entities of the most frequent types. $S_{NoWeight}$ performance

	HEAD		TAIL	
	Prec@1	**F_1**	**Prec@1**	**F_1**
$\mathbf{B_{Random}}$	0.008	0.028	0.004	0.023
$\mathbf{B_{Frequency}}$	0.244	0.302	0.298	0.322
$\mathbf{S_{NoWeight}}$	$0.542^{\dagger}$	$0.465^{\dagger}$	$0.510^{\dagger}$	$0.456^{\dagger}$
$\mathbf{S_{Weighted}}$	$0.610^{\ddagger}$	$0.521^{\ddagger}$	$0.598^{\ddagger}$	$0.522^{\ddagger}$

Table 5: Evaluation on HEAD and TAIL, 500 elements each. † indicates statistical significance over $\mathbf{B_{Frequency}}$, and ‡ over both $\mathbf{B_{Frequency}}$ and $\mathbf{S_{NoWeight}}$. Significance is measured using the Student's t-test at 95% confidence. The top type predicted by our $\mathbf{S_{Weighted}}$ method is correct about 60% of the time, while the top type predicted by the $\mathbf{B_{Frequency}}$ baseline is correct under 30% of the time.

is statistically significant above all baselines, and $S_{Weighted}$ is statistically significant over $S_{NoWeight}$ on both test sets and metrics.

5.5 Analysis

$S_{Weighted}$ was successful at recovering the correct Freebase types of many entities. For example, it finds that "Atherosclerosis" is a *medical risk factor* by connecting it to "obesity" and "diabetes," that "Supernatural" is a *TV program* and a *Netflix title* by connecting it to "House" and "30 Rock," and that "America West" is an *aircraft owner* and an *airline* by connecting it to "Etihad Airways" and "China Eastern Airlines." While precision at 1 around 60% may not be high enough yet for certain applications, it is significantly better than competing approaches, which are under 30%, and we hope that our values can serve as a non-trivial baseline on this task for future systems.

One example where $S_{Weighted}$ made some mistakes is *fictional characters*. Many *fictional characters* participate in a textual relations that make them look like *people* (e.g., "was born on"), but predicting that they belong to *people* class is incorrect. Some performance hit was also due to entity linking errors. From an assertion like "The Four Seasons is located in Hamamatsu," our entity linker (and other entity linkers we tried) prefer linking "The Four Seasons" to Vivaldi's music composition rather than the hotel chain. We are then unable to recover *music composition* type from relations like "is located in." Our algorithm relies on accurate entity linking in L, but there is a precision/recall tradeoff to consider here because it also benefits from higher coverage of entities and relations in L.

As a general reference for performance of state-of-the-art fine-grained entity classification, the FIGER system (Ling and Weld, 2012) for classifying into 112 types reported F_1 scores ranging from 0.471 to 0.693 in their experiments. It is important to note that these numbers are not directly comparable to us because they used different data, different (and fewer) types, and different evaluation methodology.

6 Discussion

This paper presented an approach for working with non-Wikipedia entities in text. Consider the following possibilities for a noun phrase in a text corpus:

Wikipedia Entity: (e.g., "Computer Science," "South America," "apple juice") - Entity linking techniques can identify and type these.

Non-Wikipedia, Non-Entity: (e.g., "strange things," "Early studies," "A link") - Our classifier from Section 4 is able to filter these.

Non-Wikipedia, Entity: (e.g., "Safflower oil," "prune juice," "Amazon UK") - We identify these as entities, then propagate semantic types to them. Our technique finds that "Safflower oil" occurs with high weight relations such as "is sometimes used to treat" and "can be substituted for," making it similar to linked entities such as "Phentermine" and "Dandelion," and then correctly predicts semantic types including *food ingredient* and *medical treatment*.

6.1 Typed Question Answering

From our set of 15 million assertions, we found and typed many non-Wikipedia entities. In *food* while Wikipedia has "crab meat," we find it is missing others such as "rabbit meat" and "goat milk." In *job titles* it has "scientist" and "lawyer," but we find it is missing "PhD student," "fashion designer," and others. We find many of the *people* and *employers* not

prominent enough for Wikipedia.

One application of this research is to increase the yield of applications such as Typed Question Answering (Buscaldi and Rosso, 2006). For example, consider the query "What *computer game* is a lot of fun?" A search for assertions matching "* is a lot of fun" in the data yields around 1,000 results such as "camping," "David Sedaris" and "Hawaii." Entity linking allows us to identify just the *computer games* in Wikipedia that match the query, such as "Civilization." However, around 400 query matches could not be entity linked. Our noun-phrase classifier filters out non-entities such as "actual play," "Just this" and "Two kids." After predicting types for the matches that did not get filtered, we find additional non-Wikipedia *computer games* that match the query, including "Cooking Dash," "Delicious Deluxe" and "Slingo Supreme."

7 Future Work

An area we are continuing to improve the system on is *textual ambiguity*. For example, an unlinkable noun phrase might simultaneously be the name of a *film*, a *car*, and a *person*. Instead of outputting that the noun phrase holds all of those types, a stronger output would be to realize that the noun phrase is ambiguous, determine how many senses it has, and determine which sense is being referred to in each instance. We have ideas for how to detect ambiguous entities using mutual exclusion (Carlson, 2010) and functional relations. For example, if we predict that a noun phrase has *film* and *car* types but we also observe in our linked instances that these types are mutually exclusive, then this is good evidence that the noun phrase refers to multiple terms.

We also plan to continue improving our techniques, as there is still plenty of room for improvement on both subtasks. For detecting new entities, we are interested in seeing if timestamped Twitter data could be analyzed to increase both recall and precision. For predicting semantic types, (Kozareva et al., 2011) proposed additional techniques which we have not fully explored. Also, we can incorporate additional signals such as shared term heads when they are available, in order to help find terms that are likely to share types. Last, we would like to feed back our system output to improve system performance. For example, non-entity noun phrases that make it to the typing step might lead to particular predicted type distributions that indicate an error occurred earlier in the process.

8 Conclusion

In this paper we showed that while entity linking cannot link to entities outside of Wikipedia, once a large text corpus has been entity linked, the presence and content of the existing links can be leveraged to help detect and semantically type the non-Wikipedia entities as well. We designed techniques for detecting whether unlinkable noun phrases are entities, and if they are, then propagating semantic types to them from the linked entities. In our evaluations, we showed that our techniques achieve statistically significant improvement over baseline methods.

Our research here takes initial steps toward a future where the vast universe of entities that are not prominent enough to include in manually-authored knowledge bases is analyzed automatically instead of being left behind.

Acknowledgements

We thank Stephen Soderland, Xiao Ling, and the three anonymous reviewers for their helpful feedback on earlier drafts. This research was supported in part by NSF grant IIS-0803481, ONR grant N00014-08-1-0431, and DARPA contract FA8750-09-C-0179, and carried out at the University of Washington's Turing Center.

References

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association of Computational Linguistics (EACL)*.

Davide Buscaldi and Paolo Rosso. 2006. Mining knowledge from wikipedia for the question answering task. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*.

Andrew Carlson. 2010. *Coupled Semi-Supervised Learning*. Ph.D. thesis, Carnegie Mellon University.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP*.

Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. 2010. Entity disambiguation for knowledge base population. In *Proceedings of COLING*.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the Web: An experimental study. In *Artificial Intelligence*.

Oren Etzioni, Michele Banko, and Michael J. Cafarella. 2006. Machine Reading. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*.

James A. Evans and Jacob G. Foster. 2011. Metaknowledge. In *Science*.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2009. Scaling Wikipedia-based named entity disambiguation to arbitrary Web text. In *IJCAI-09 Workshop on User-contributed Knowledge and Artificial Intelligence (WikiAI09)*.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of EMNLP*.

Paolo Ferragina and Ugo Scaiella. 2010. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of CIKM*.

Eduard Hovy, Zornitsa Kozareva, and Ellen Riloff. 2009. Toward completeness in concept extraction and classification. In *Proceedings of EMNLP*.

Zornitsa Kozareva, Konstantin Voevodski, and Shang-Hua Teng. 2011. Class label enhancement via related instances. In *Proceedings of EMNLP*.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of wikipedia entities in text. In *Proceedings of KDD*.

Changki Lee, Yi-Gyu Hwang, and Myung-Gil Jang. 2007. Fine-grained named entity recognition and relation extraction for question answering. In *Proceedings of SIGIR*.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva P. Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, and Steven Pinker. 2010. Quantitative analysis of culture using millions of digitized books. In *Science*.

David Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17h ACM International Conference on Information and Knowledge Management (CIKM)*.

Marius Paşca. 2004. Acquisition of categorized named entities for web search. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*.

Patrick Pantel and Deepak Ravichandran. 2004. Automatically labeling semantic classes. In *Proceedings of HLT-NAACL*.

Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of EMNLP*.

Danuta Ploch. 2011. Exploring entity relations for named entity disambiguation. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*.

Willard Van Orman Quine. 1948. On what there is. In *Review of Metaphysics*.

Altaf Rahman and Vincent Ng. 2010. Inducing fine-grained semantic classes via hierarchical and collective classification. In *Proceedings of COLING*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*.

Satoshi Sekine and Chikashi Nobata. 2004. Definition, dictionaries and tagger for extended named entity hierarchy. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*.

Partha Pratim Talukdar and Fernando Pereira. 2010. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*.

Partha Pratim Talukdar, Joseph Reisinger, Marius Paşca, Deepak Ravichandran, Rahul Bhagat, and Fernando Pereira. 2008. Weakly supervised acquisition of labeled class instances using graph random walks. In *Proceedings of EMNLP*.

Partha Pratim Talukdar, Derry Tanti Wijaya, and Tom Mitchell. 2012. Coupled temporal scoping of relational facts. In *Proceedings of WSDM*.

Chi Wang, Kaushik Chakrabarti, Tao Cheng, and Surajit Chaudhuri. 2012. Targeted disambiguation of ad-hoc, homogeneous sets of named entities. In *Proceedings of the 21st International World Wide Web Conference (WWW)*.

Derry Tanti Wijaya and Reyyan Yeniterzi. 2011. Understanding semantic changes of words over centuries. In *Workshop on Detecting and Exploiting Cultural Diversity on the Social Web*.