Semantic Lexicon Induction from Twitter with Pattern Relatedness and Flexible Term Length

Ashequl Qadir
University of Utah
50 S Central Campus Drive
Salt Lake City, Utah 84112
asheq@cs.utah.edu

Pablo N. Mendes, Daniel Gruhl and Neal Lewis
IBM Research Almaden
650 Harry Road
San Jose, California 95120
pnmendes,dgruhl,nrlewis@us.ibm.com

Abstract

With the rise of social media, learning from informal text has become increasingly important. We present a novel semantic lexicon induction approach that is able to learn new vocabulary from social media. Our method is robust to the idiosyncrasies of informal and open-domain text corpora. Unlike previous work, it does not impose restrictions on the lexical features of candidate terms – e.g. by restricting entries to nouns or noun phrases – while still being able to accurately learn multiword phrases of variable length. Starting with a few seed terms for a semantic category, our method first explores the context around seed terms in a corpus, and identifies context patterns that are relevant to the category. These patterns are used to extract candidate terms – i.e. multiword segments that are further analyzed to ensure meaningful term boundary segmentation. We show that our approach is able to learn high quality semantic lexicons from informally written social media text of Twitter, and can achieve accuracy as high as 92% in the top 100 learned category members.

Introduction

For many Natural Language Processing tasks (e.g., Information Extraction, Sentiment Analysis), understanding the meaning of a phrase is profoundly important. Semantic lexicons allow us to associate an "is-a" relation between a semantic category and its members (e.g., *"burger"* is a FOOD and *"football"* is a SPORT), and sits at the heart of many business analytic tools. Although there exist knowledge resources like WordNet (Miller 1995), building such resources require extensive human effort, making it difficult to keep up with neologism (e.g., *"cronut"*, a FOOD term, did not exist before 2010). Additionally, increasing popularity of social media over the last decade has given rise to new challenges when knowledge needs to be acquired from informally written text. For example, in Twitter microblogging platform, it is common to use informal or abbreviated words like *"bball"* to mean *"basketball"* which is a SPORT, or *"cod"* to refer to *"Call of Duty"*, which is a VIDEO GAME.

One of the prominent characteristics of social media text is informal writing style and often unconventional grammar. General-purpose language parsers and parts-of-speech taggers have been found to perform poorly when applied on tweets (Gimpel et al. 2011; Foster et al. 2011). Special-purpose language parsers exist (e.g., (McClosky and Charniak 2008)), but do not often adapt well to other text genres, domains or natural languages. As a result, lexicon induction approaches that are dependent on lexico-syntactic context patterns (e.g., (Riloff and Jones 1999; Thelen and Riloff 2002)) are less flexible and more susceptible to upstream errors. Moreover, previous work that limits their learning scopes to single nouns or head nouns (Thelen and Riloff 2002), inherently limits their effectiveness for multiword terms. For example, *"nugget"* is not generally considered a FOOD item, but *"chicken nugget"* clearly is. Likewise, while *"game"* is considered a SPORT EVENT, other phrases like *"blame game"* or *"waiting game"* are not. Many terms may also consist of other parts-of-speech (e.g., "falls asleep frequently" is a MEDICAL SYMPTOM). This emphasizes the need for strategies that are able to discover meaningful multiword terms, and are not confined within the scope of learning terms with specific parts-of-speech only.

Other approaches have used N-gram based context patterns and learned multiword terms. But many limited their scopes to either title-case N-grams, bigrams that already exist in WordNet (Murphy and Curran 2007) or multiword lexemes from Wikipedia article names (Kolb 2008). As a result, informally written longer phrases like *"hide n go seek"*, which refers to a CHILDREN'S GAME or *"mac and cheese"* referring to a FOOD DISH are beyond their learning scopes.

We present a novel semantic lexicon induction approach that is more flexible and accurate than previous work. Our approach is rooted on the ability to extract and prioritize N-gram context patterns that are *semantically related* to the categories. It does not depend on syntactic dependencies in a sentence, and is able to discover and classify multiword terms without restrictions on their lexical features. Curran (2004) also observed that not all patterns are equally reliable. For example *"I ate ()"*[1] in *"I ate pizza"* is a more reliable pattern for learning FOOD terms, than the pattern*"I want ()"* in *"I want pizza"*. But unlike Curran (2004) who used only verb contexts (e.g., *"wear"* for CLOTHES) and required a target word to be syntactically tied as the subject, direct object or indirect object to a verb, we hypoth-

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1] Henceforth, we use this construction to indicate context patterns where "()" indicates an empty slot that can be filled by a term.

esize that any semantically related word regardless of its parts-of-speech and syntactic ties is capable of providing dependable context when the same context is also found with known category terms. Budanitsky and Hirst (2006) defined semantically related entities as either semantically similar (e.g., *"bank"*-*"trust company"*), or dissimilar (e.g., *"paper"*-*"pencil"*) entities, that are associated with each other by any functional, meronymy, antonymy, frequent association, etc. relations. We use this definition for words instead of entities (e.g., *"restaurant"*, *"baked"*, *"delicious"*, *"chef"*, etc. for learning FOOD terms).

The second observation we make is that the patterns that appear on the left, right or around a potential category term are often also frequent with other members of the same category. For example, we would expect to find contexts such as *"I ate ()"* and *"() were delicious"* with *"chicken wings"* and *"pizza slices"* alike. But while *"I ate (chicken wings at) ..."* will likely be found in a sentence, *"(chicken wings at) were delicious"* will be unlikely, and this can help us eliminate "chicken wings at" as a FOOD term. Based on this hypothesis, we introduce a method to select candidate category terms that have more probable term boundaries than others.

We show that our novel contribution of using the semantically related words to find and rank candidates along with the method for term selection with suitable boundary yield accuracy as high as 92% in the top 100 learned category members, and is able to learn many multiword terms that were beyond the scope of previous work.

Related Work

One of the important considerations in lexicon induction[2] research is the discovery of new candidate terms. Non-iterative methods experimented with noun phrase chunks and distributional similarity (Pantel et al. 2009), exploited list structures in text (Sarmento et al. 2007) and web pages (Wang and Cohen 2007), and also used N-gram context patterns from web search query logs (Paşca 2007).

Another line of work which uses iterative bootstrapped learning techniques, starts with a few seed words for a semantic category and iteratively adds new terms to the learned lists. For discovering new candidates, these methods have considered nouns that appeared near seeds (Riloff and Shepherd 1997) or utilized compound nouns and other syntactic constructions (Roark and Charniak 1998). Other works exploited syntactic heuristics (Phillips and Riloff 2002), lexico-syntactic patterns (Riloff and Jones 1999; Thelen and Riloff 2002), weighted context N-grams of seeds (Murphy and Curran 2007; McIntosh and Curran 2008), pre-designed and automatically learned context patterns (Pasca 2004), domain-specific extraction patterns (Etzioni et al. 2005) and doubly anchored hyponym patterns (Kozareva, Riloff, and Hovy 2008). Although many focused solely on learning single nouns (e.g., (Thelen and Riloff 2002)), some approaches also learned multiword terms using pre-designed context patterns and exploiting web as a corpus (e.g., (Kozareva, Riloff, and Hovy 2008; Paşca 2007)) and list structures (e.g., (Sarmento et al. 2007; Wang and Cohen 2007)) in Wikipedia or web pages. In other cases, researchers also learned multiword terms using title-case N-grams or bigrams that already exist in WordNet (Murphy and Curran 2007), or multiword lexemes that exist in other resources or can be found as Wikipedia articles names (Kolb 2008). These approaches primarily benefited from the vast information in the web, or special structures expected in corpus. As a result, these approaches may not directly apply to tweets which are short in nature, and are well known for informal writing practices.

A well acknowledged concern about the iterative learning methods is that noisy inclusion of new category members affects successive iterations and may result in semantic drift. Thelen and Riloff (2002) learned multiple categories simultaneously to restrict candidate term space of each category. Murphy and Curran (2007) used mutual exclusion bootstrapping to minimize semantic drift for both terms and contexts. McIntosh and Curran (2009) reduced semantic drift with bagging and distributional similarity. McIntosh (2010) introduced negative categories when semantic drift has occurred. Carlson et al. (2009) simultaneously learned classifiers constrained with entity relations. Qadir and Riloff (2012) designed an ensemble of component learning systems to learn only the category members that have consensus of the components. Although iterative learning is beyond the scope of our current research, it is a potential extension to our approach, and we consider it as a promising future work direction.

Kolb (2008) presented DISCO, a distributional similarity based method to determine semantic similarity between a pair of words using second order word co-occurrences, and semantic relatedness, using first order word co-occurrences.[3] Our work differs as we use semantically related words for the task of semantic lexicon induction. More recently, De Benedictis, Faralli, and Navigli (2013) presented GlassBoot, a minimally-supervised bootstrapping algorithm that acquires domain glossaries from the web, and exploits learned glosses to extract term hypernyms. Their method is web specific and leverages html block elements to determine boundaries for term glosses. The method also inherently expects that the gloss of a term is explicitly written in the web by someone. As such characteristics are unlikely to be frequent for tweets, it may not directly apply. Additionally, one may not always expect to find explicitly written glosses for informal terms with creative spelling, as can be commonly found in Twitter short messages.

While many research used domain specific text corpora to expand their domain specific lexicons, others used cross-domain text such as Wikipedia or used the Web as a corpus. To the best of our knowledge, we are the first to learn semantic lexicons from tweets where many category terms are informally expressed, and so are their contexts where the terms appear. Coden et al. (2012) used N-gram context patterns and learned drug lexicons from clinical progress notes, which are also expected to contain text that does not always

[2] Also sometimes addressed as *Set Expansion*.

[3] Kolb (2008) defined *semantically similar* words as words that can be substituted in the same context, which they mentioned must not be true for *semantically related* words.

follow formal English grammar, contains misspellings and abbreviations, much like tweets. However, our approach is substantially different from previous work as we use semantically related words to determine which context patterns are more reliable for better candidate discovery. Additionally, we do not impose restriction on lexical features (e.g., parts-of-speech) of terms, and present a novel method to automatically select candidates with suitable term boundaries based only on context patterns learned from seed terms.

Semantic Lexicon Induction

Our goal is: given a semantic category C and a set of seed terms $s \in S_c$ such that every s is associated with C with an *"is a"* relation, we expand S_c by learning new terms $t \in T$ where T is the set of all terms in a corpus, such that t is also associated with C with an *"is a"* relation. Each of s and t is comprised of a sequence of words $w_{1 \ldots n}$ where n is not pre-determined. That is, n is a parameter to the algorithm so that the algorithm can learn terms of size 1, 2, 3, up to length n. Although in our experience $n = 6$ has been sufficient in most cases, it can be set arbitrarily high.

Semantic Categories

For this research, we experimented with 3 semantic categories[4] that we expect people to generally talk about in Twitter among other topics. These 3 categories are:

Food & Drinks: General food terms (e.g., *pizza*, *cake*) and drinks (e.g., *water*, *wine*). We also include food ingredients (e.g., *salt*), food nutrients (e.g., *carbohydrates*, *proteins*) as well as common food metonyms by nationality (e.g., *mexican*, *chinese*) and brand names (e.g., *nutella*, *cheetos*). We do not include possible metonymic terms that are more ambiguous in the context of FOOD & DRINKS, such as, restaurant names (e.g., *McDonalds, Taco Bell*)[5].

Games & Sports : Indoor and outdoor sports (e.g., *soccer*, *baseball*, *table tennis*), as well as different types of games such as board games (e.g., *chess*), card games (e.g., *poker*, *uno*), video games (e.g., *tomb raider*, *call of duty*), etc. We also include sports events (e.g., *olympics*, *marathon*), but exclude possible metonyms that are ambiguous in the context of SPORTS & GAMES such as league names (e.g., *NFL*) and general outdoor activities that are not normally considered as sports (e.g., *camping*, *hiking*).

Vehicles: General vehicle terms (e.g., *car*, *bus*, *train*) as well as automobile brands and manufacturing companies (e.g., *toyota*, *honda*). We also include well known transportation systems that are commonly used to refer to the vehicle itself (e.g., *subway*, *metro*).

For each of our categories, we manually selected 10 seed terms that are frequent in our corpus. Table 1 presents the seed terms we selected for these categories.

FOOD & DRINKS	GAMES & SPORTS	VEHICLES
food	soccer	car
water	cricket	truck
wine	basketball	vehicle
drinks	hockey	vehicles
pizza	tennis	nissan
beverage	volleyball	van
cheese	table tennis	ford
fish	baseball	chevy
chicken	american football	honda
chocolate	rugby	suv

Table 1: Seed terms for categories.

Corpus Description & Pre-processing

For learning our semantic lexicons from informally written text, we use tweets as our dataset. Tweets are short messages with a maximum length of 140 characters, supported by the Twitter microblogging platform. Tweets are well known for informal grammar, abbreviated expressions, and misspellings, which make them challenging to apply well known Natural Language Processing tools. For this research, we collected 114 million English tweets published in Twitter during February and March, 2013, using Twitter 10% decahose stream. For pre-processing, we tokenized each tweet using an in-house tokenizer and normalized with respect to case.

Context Pattern Selection

For each seed term $s \in S_c$ for semantic category C, we first extract all N-gram context patterns containing up to 6 words, and store them in our pattern pool P_c. For each pattern $p \in P_c$, we calculate pattern confidence from the percentage of unique seed terms that a pattern matches[6].

$$\text{confidence}(p) = \frac{\text{Num. unique seed terms matched by } p}{\text{Num. all terms matched by } p}$$

We then remove any $p \in P_c$ that has a confidence threshold lower than 10^{-6} to limit the initial pattern space. We set this threshold arbitrarily low so as to take into account the majority of the seed matching patterns and discard patterns that rarely appear with a seed term.

The objective of context pattern selection is to determine which context patterns are more reliable for learning category terms. Our hypothesis is that when a context pattern contains a semantically related word, it is more likely to discover better category terms. For example, consider the following 3 patterns[7]: *"I played ()"*, *"the () stadium was full", and "I love watching ()"*. Because *"played"* and *"stadium"* are semantically related to SPORTS & GAMES, intuitively, the first two patterns are more likely to match terms like *"football"*, *"baseball"*, etc, whereas the third pattern can match a wide range of terms like *"movies"*, *"birds"*,

[4] The selection of these 3 semantic categories was driven by business need, and we leave other more general or specific semantic categories for future work.

[5] It is common to say *"I bought nutella at the grocery"* to mean *"nutella chocolate spread"*. Similarly, *"I am having McDonalds for lunch"* is also common. But *"McDonalds"* in *"I am eating at McDonalds"* will not be a metonymy reference.

[6] A "match" happens when a term is found in the placeholder position "()" of a context pattern.

[7] Note that actual context patterns are not always as well formed in tweets.

"football", etc. It is also important to note that having a semantically related term does not always guarantee a relevant candidate. For example, *"I played (violin)"* and *"the (Liverpool) stadium was full"* are both likely matches. However, we expect that the patterns that contain a semantically related term will substantially restrict the candidate space.

Finding semantically related terms is a different research problem in itself, and is out of our research scope. In principle, any semantic similarity tool should work, and in this research we do not intend to compare how well different semantic similarity tools perform. We use a freely available tool, DISCO (Kolb 2008), to provide us with semantically related words, generated from first order word collocations. For each seed term $s \in S_c$ for our category C, we obtain the first 200 semantically related words using DISCO. We then take a union of the generated words, and create the set of semantically related words W_c for category C. For each pattern $p \in P_c$, we then keep p in P_c only if p contains any $w \in W_c$.

Candidate Term Discovery

For the next stage of our learning, we search our entire tweet corpus to find any term that is matched by a pattern $p \in P_c$, and extract all terms $t_{w1...wn}$. As the resulting set of candidates is too large to consider all of them for our subsequent stages of lexicon induction, we select a smaller subset of candidates within our budget that we want to further evaluate with more expensive computations. To select this smaller subset, we score all the candidates by counting the number of unique patterns in P_c that extract them, and rank the candidates based on this score in descending order. We then take the top 2000 candidates as our initial set of candidates T_c for category C.

Candidate Term Boundary

One of the novel contributions of our research is to select candidates with more appropriate term boundaries. To illustrate the challenge, consider the context pattern *"I ate ()"*. While at a first glance, it may seem like a reliable context pattern to extract FOOD terms, in reality, this context pattern may potentially extract terms like *"sandwich"*, *"sandwich today"*, *"sandwich today at"*, *"sandwich today at lunch"*, *"sandwich today at lunch with"*, *"sandwich today at lunch with ketchup"*, etc. Moreover, all these extracted terms may also potentially match many other semantically related patterns such as *"I was eating ()"*, *"it was a delicious ()"*, *"Went to McDonalds and had a ()"*, etc.

Even when terms are surrounded by context patterns from two sides, with indefinite or large term lengths, the patterns are not guaranteed to extract terms with suitable boundaries. For example, the context pattern *"I was eating () yesterday"* may potentially extract candidates such as *"I was eating (sandwich with a friend) yesterday"* or *"I was eating (sandwich when my friend called) yesterday"*, etc. This issue is less likely to occur when noun phrase chunks or only head nouns are used as can be seen in related work (e.g., (Thelen and Riloff 2002; Pantel et al. 2009)). However, since such linguistic tools work less reliably for tweets (Gimpel et al. 2011; Foster et al. 2011), we propose the following novel solution in this research.

We classify context patterns based on their position relative to term matches: rightmost-patterns include patterns for which seeds match only on the right-hand side (e.g., , *"I was eating ()"*), leftmost-patterns are those where seeds match only on the left side (e.g., *"() for dinner"*), and middle patterns have text on both sides of matching seeds (e.g., *"eating () for dinner"*). Our hypothesis is that term candidates with good term boundaries will match left, right and middle patterns, while terms with unsuitable boundaries will not always match all three. For example, a non-ideal term such as *"sandwich today with"* will match left-patterns like *"I ate ()"*, *"I was eating ()"* or *"It was a delicious ()"*, but it will be less likely to also match right-patterns like *"() for dinner"*, *"() with cheese"*, etc.

To validate this hypothesis, we built a pattern pool (length 1 to 6 words) from a random selection of 10,000 tweets for each seed and candidate term. We then create 3 pattern subsets $P_{c,t,l}$, $P_{c,t,r}$ and $P_{c,t,m}$, where l, r or m indicates the *leftmost*, *rightmost* or *middle* position in a pattern where the candidate term t is matched. Similarly, we also create $P_{c,S,l}$, $P_{c,S,r}$ and $P_{c,S,m}$, depending upon the position in a pattern where a seed term $s \in S_c$ is matched.

We rate how well a candidate's boundary "fits" by looking for respective seed matching patterns that also match the candidate to the left, right and middle. The position score PS, at positions $pos \in \{l, r, m\}$ for a candidate term t is computed by the number of patterns that match both t and any given seed in S. Scores are compressed in logspace to make them less sensitive to small differences in counts.

$$PS_{c,t}(pos) = log(\left|P_{c,S,pos} \cap P_{c,t,pos}\right|)$$

Finally, the term boundary score $TBS_{c,t}$ is computed by taking the harmonic mean of the position scores. The intuition is that a candidate with suitable term boundary should have high scores for all three positions, whereas candidates with fewer patterns in common with the seeds at any of the positions will get a lower score.

$$TBS_{c,t} = \frac{3 * PS_{c,t}(l) * PS_{c,t}(m) * PS_{c,t}(r)}{PS_{c,t}(l) * PS_{c,t}(m) + PS_{c,t}(l) * PS_{c,t}(r) + PS_{c,t}(m) * PS_{c,t}(r)}$$

We keep only the candidates that pass a minimum score threshold. We use the average term boundary score for all $t \in T_c$ to only keep the candidates that has a TBS greater than the average.

Candidate Term Ranking

The last stage of our lexicon induction is the candidate ranking. To rank the candidate terms, we use only the most reliable patterns that extract them. Although we initially selected the patterns in P_c by exploiting semantically related words (top 200 words for each seed $s \in S$) obtained using DISCO (Kolb 2008), the generated word sets are far from perfect, and do not always guarantee semantically related words. To overcome this issue, we only keep a smaller subset of the patterns from P_c that we feel confident about. To do that, first we score each context pattern $p \in P_c$ by counting how many unique seeds from S are extracted by each pattern p. Then we rank the patterns by this score in descending order, and keep only the top 20% of the patterns ranked by the score.[8]

To rank the candidates, our hypothesis is that if a candidate term t actually belongs to the semantic category C, then among all the patterns that extract t, the percentage of semantically related reliable unique patterns will be much higher. To approximate this ratio for t, we use our larger pattern pool $P^*_{c,t}$ created from the randomly selected 10,000 tweets for each candidate term $t \in T_c$. We then score t by counting how many of the patterns in P_c occurs among the patterns in $P^*_{c,t}$ and then taking the ratio.

$$score(t) = \frac{\text{num patterns in } P_c \text{ that appears in } P^*_{c,t}}{\text{num patterns in } P^*_{c,t}}$$

Finally, we create a ranked list of the candidate terms for category C based on this calculated score, and take the top 200 terms for evaluation. Table 2 presents examples of the category terms taken from top 30 lexicon entries generated by our approach.

FOOD & DRINKS	SPORTS & GAMES	VEHICLE
cornbread	basket ball	golf cart
grilled chicken	field hockey	jeep
mac n cheese	flag football	snowmobile
chicken wings	water polo	first car
asparagus	vball	lorry
hotdogs	quidditch	motorbike
tofu	footie	minivan
hot wings	high school football	school bus
lucky charms	frisbee	race car
sausages	ice hockey	moms car
porridge	scrabble	hummer
buffalo chicken	snooker	tractor
ramen noodles	hide and seek	motorcycle
chocolate chip cookies	dodgeball	limo
fried rice	netball	lexus
pizza rolls	guitar hero	ferrari
chocolate covered-strawberries	high school-basketball	motor-vehicle
grits	beer pong	benz
hot dogs	club penguin	bentley
crawfish	paintball	porsche

Table 2: Example of category terms taken from the top 30 learned terms.

Evaluation

Baselines

To compare the quality of our lexicons with previous work, we used two widely cited approaches: DISCO (Kolb 2008) and BASILISK (Thelen and Riloff 2002). In order to directly evaluate scoring functions and control for potential differences in implementation details, we provided both systems with the same list of 2000 candidates T_c, and used the systems to score and rank the list.

For our first comparison, we use DISCO, which allows to retrieve semantic similarity between arbitrary words using methods that leverages distributional similarity. Kolb (2008) showed that second order word collocations improve over first order for finding semantically similar terms. Therefore, for each term t in T_c, we retrieve the similarity scores from DISCO using second order word collocations with our category names[9], and keep the highest similarity score for each term $t \in T_c$. We then rank the list with this score and take the top 200 terms. For a second comparison, we retrieve similarity scores with the seed words of each category. We refer to these lists as *"Disco SimCat"* and *"Disco SimSeed"*.

For our next comparison, we use BASILISK – an iterative bootstrapping algorithm that learns a few words per iteration to incrementally expand categories. We first provided BASILISK with all context patterns in our $P{*}_{c,t}$ for all candidate term $t \in T_c$ for category C, which we previously generated by collecting 10,000 tweets for each $t \in T_c$. Since BASILISK only learns head nouns, we presented to BASILISK all multiword terms of our candidate list as single token terms by replacing whitespaces with underscore. We then ran BASILISK in multi-category mode with improved conflict resolution for a single iteration.

It is important to mention that BASILISK was originally designed as a bootstrapping algorithm to iteratively build lexicons, and using BASILISK for a single iteration limits its capabilities. However, as iterative learning is beyond the scope of this research, for the sake of comparison we only compare with a single iteration of BASILISK. By design, to limit candidate space, BASILISK chooses only 20 seed-matching patterns in the first iteration, and only evaluates candidate terms that are matched by these 20 patterns. As a result, the generated list of candidates in the single iteration from BASILISK was very small compared to ours – they only contained 30, 18 and 17 terms for our 3 categories. Therefore, we also reimplemented and evaluated *AvgLog* and *Diff*, BASILISK's two candidate scoring function on the candidates generated by our approach.

$$\text{AvgLog}(t) = \frac{\sum_p \log_2(F_p+1)}{|P^*_{c,t}|}$$

Here, $|P^*_{c,t}|$ is the number of patterns that extract term t for category C, and F_p is the distinct number of seeds extracted by each pattern $p \in p^*_{c,t}$.

$$\text{Diff}(t) = \text{AvgLog}(t, c) - \max_{c \neq c_o} \text{AvgLog}(t, c_o)$$

We then rank the candidates based on both of these scores separately and take the top 200 terms for each category. We refer to these learned lexicons as *Basilisk 1-iter*, *Basilisk AvgLog*, *Basilisk Diff*.

Gold Standard

For each semantic category, we combined the 200 candidate terms generated by each of the methods, and provided them to two annotators without indication to which method produced which candidate. The annotators were then given clear annotation guidelines along with category definitions

[8] This threshold is determined from empirical observation of the ranked context patterns in our study.

[9] We use both singular and plural forms (e.g., *"vehicle"* and *"vehicles"* for the category VEHICLE)

	Food & Drinks				Sports & Games				Vehicle			
Lexicon Sizes	25	50	75	100	25	50	75	100	25	50	75	100
Basilisk												
Basilisk 1-iter	.75	.63 (30)			.78 (18)				.65 (17)			
Basilisk AvgLog	.56	.70	.77	.76	.48	.66	.73	.75	.04	.10	.12	.19
Basilisk Diff	.88	.86	.83	.86	.68	.74	.79	.76	.04	.20	.28	.35
Disco												
Disco SimCat	.80	.76	.79	.78	.88	.70	.56	.45	.84	.74	.56	.46
Disco SimSeed	.76	.78	.81	.84	.88	.64	.49	.41	.80	.68	.57	.50
RelW												
RelW	.64	.74	.79	.79	.52	.74	.79	.80	.04	.12	.27	.35
RelWBound	**1.00**	**.98**	**.95**	**.92**	**.96**	**.92**	**.88**	**.83**	**.88**	**.78**	**.67**	**.57**

Table 3: Accuracy of the induced lexicons up to top 100 terms.

and examples, and were asked to assign category membership to each term. The annotators were also instructed to only assign a semantic category to a term if the term had a right term boundary. Cohen's Kappa (κ) was 0.86, indicating high agreement between the annotators. We then used the annotated lists as our gold standard for evaluating the generated lexicons.

Results

To compare the lexicons, we use accuracy as our evaluation metric where a term is considered "accurate" only if it has been assigned the right semantic class and it has the right term boundary. Table 3 shows the accuracy of the lexicons up to first 100 terms. *Basilisk 1-iter*, although very short, shows that it learned terms in all three categories with reasonable accuracy, but the accuracy is still much lower compared to the other systems. Comparatively, *Basilisk AvgLog* ranked more terms with good accuracy for Food & Drinks and Sports & Games. *Basilisk Diff* performed substantially better than *Basilisk AvgLog* in all three categories. Both methods performed poorly for Vehicle. An analysis of the lexicons suggested that many terms were selected earlier in the lexicon that did not have the right term boundaries. Some of the example terms with incorrect term boundaries that *Basilisk Diff* selected are: *"want some chicken"*, *"chocolate right now"* under Food & Drinks and *"broken legs after motorcycle"*, *"car lol"*, etc. under Vehicle.

Next, the two Disco lexicons: *Disco SimCat* and *Disco SimSeed* worked reasonably well for Food & Drinks, but only at the earlier section of the lexicons for Sports & Games and Vehicle. We found that the Disco lexicons failed to recognize many informal terms people use to refer to sports in tweets (e.g., *"college ball"*) and terms that do not have Wikipedia articles of their own (e.g., *"race car"*).

In the the last two rows of Table 3, the *RelW* row refers to our approach that ranks the candidates using semantically related patterns, but do not use the term boundary detection method. The accuracy is close to *Basilisk Diff* and *Disco SimSeed* for Food & Drinks and close to *Basilisk Diff* for the other two categories. Notably, it did not perform well at the beginning as many terms were included in the lexicons in the top positions that did not have the right term boundaries. This demonstrates the necessity of a method that is able select terms with meaningful boundaries.

The last row in Table 3 presents the accuracy of our learned lexicons, *RelWBound*, that uses the selection of terms with suitable boundary, and also ranks the terms using semantically related patterns. We find that in all three categories, *RelWBound* was able to consistently learn terms with much higher accuracy than all the other systems. An important point to note is that for the first 100 terms, the accuracy of *RelW* kept increasing as the lexicon sizes increased, but the accuracy for *RelWBound* started decreasing slowly. Further analysis of the learned lexicons revealed that as *RelWBound* learned many of the right category members sooner, the scope of learning more correct terms from the initial candidate list became limited. Also, the method for discarding terms with non-ideal boundaries is not perfect, and *RelWBound* occasionally discarded a few terms that indeed had the right term boundaries.

We additionally looked into how many of the learned terms do not exist in WordNet (Miller 1995), which is a well known resource for semantic knowledge. We found that among the correct terms from the first 100 that *RelWBound* learned for each category, 21.74% of the terms in Food & Drinks, 48.49% of the terms in Sports & Games and 24.56% of the terms in Vehicle do not exist in WordNet. The reason can be attributed to informal mentions (e.g., *"mac n cheese"*, *"footy"*, *"lax"*), multiword terms not present in wordnet (e.g., *"pizza rolls"*, *"beer pong"*), brand names (e.g., *"kool aid"*, *"bmw"*, *"ferrari"*) and recently created video games (e.g., *"candy crush"*, *"temple run"*) and sports (e.g., *"quidditch"*).

Finally, Figure 1 shows the growth rate of the lexicons up to first 200 ranked terms, by method from each system demonstrating higher accuracy early on. We see that all methods had a consistently high expansion rate for Food. This is not surprising because people frequently talk about Food in Twitter. The decreased expansion rate of *Disco SimCat* in the other two categories can be attributed to many informal terms that *Disco* failed to recognize. As *Basilisk Diff* was given all the candidate terms transformed into single tokens, it eventually found many category terms, but not until later in the lexicons. On the contrary, *RelWBound*

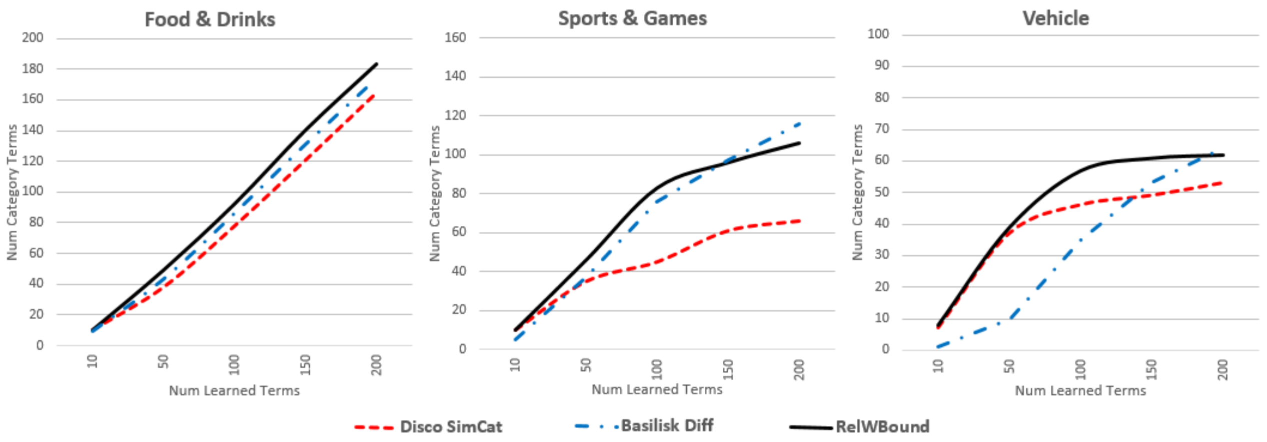

Figure 1: Lexicon Growth Rate Comparison

found more terms with ideal boundaries early on and had a steeper expansion rate than the other systems. As good accuracy and faster expansion rate are essential for iterative lexicon induction algorithms to steer the learning in the right direction as early as possible, this also makes our method promising for iterative lexicon induction, which we leave as our future work direction.

Conclusion

We presented a semantic lexicon induction method that is general enough to be able to learn semantic lexicons even from the informal text of Twitter. We demonstrated that our novel contribution of using semantically related words to select context patterns is reliable for discovering and ranking category members. Our approach did not explicitly impose pre-defined term boundary restriction, rather discovered category members with suitable term boundaries by comparing their context patterns with known category members. As future work direction, we will use the approach to learn lexicons in iterative learning framework.

Acknowledgments.

The authors would like to thank Professor Ellen Riloff (University of Utah), and the AAAI reviewers for their valuable feedback.

References

Budanitsky, A., and Hirst, G. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Comput. Linguist.* 32(1):13–47.

Carlson, A.; Betteridge, J.; Hruschka, Jr., E. R.; and Mitchell, T. M. 2009. Coupling semi-supervised learning of categories and relations. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, SemiSupLearn '09, 1–9. Stroudsburg, PA, USA: Association for Computational Linguistics.

Coden, A.; Gruhl, D.; Lewis, N.; Tanenblatt, M.; and Terdiman, J. 2012. Spot the drug! an unsupervised pattern matching method to extract drug names from very large clinical corpora. In *Healthcare Informatics, Imaging and Systems Biology (HISB), 2012 IEEE Second International Conference on*, 33–39.

Curran, J. R.; Murphy, T.; and Scholz, B. 2007. Minimising semantic drift with mutual exclusion bootstrapping. *Proceedings of the Conference of the Pacific Association for Computational Linguistics* 172–180.

Curran, J. R. 2004. *From Distributional to Semantic Similarity*. Ph.D. Dissertation, University of Edinburgh, Edinburgh, UK.

De Benedictis, F.; Faralli, S.; and Navigli, R. 2013. Glossboot: Bootstrapping multilingual domain glossaries from the web. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 528–538. Sofia, Bulgaria: Association for Computational Linguistics.

Etzioni, O.; Cafarella, M.; Downey, D.; Popescu, A.-M.; Shaked, T.; Soderland, S.; Weld, D. S.; and Yates, A. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.* 165(1):91–134.

Foster, J.; etinoglu, .; Wagner, J.; Roux, J. L.; Hogan, S.; Nivre, J.; Hogan, D.; and van Genabith, J. 2011. #hardtoparse: Pos tagging and parsing the twitterverse. In *Analyzing Microtext*, volume WS-11-05 of *AAAI Workshops*. AAAI.

Gimpel, K.; Schneider, N.; O'Connor, B.; Das, D.; Mills, D.; Eisenstein, J.; Heilman, M.; Yogatama, D.; Flanigan, J.; and Smith, N. A. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, 42–47. Stroudsburg, PA, USA: Association for Computational Linguistics.

Kolb, P. 2008. DISCO: A Multilingual Database of Distributionally Similar Words. In Storrer, A.; Geyken, A.; Siebert, A.; and Würzner, K.-M., eds., *KONVENS 2008 –*

Ergänzungsband: Textressourcen und lexikalisches Wissen, 37–44.

Kozareva, Z.; Riloff, E.; and Hovy, E. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of ACL-08: HLT*, 1048–1056. Columbus, Ohio: Association for Computational Linguistics.

McClosky, D., and Charniak, E. 2008. Self-training for biomedical parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, HLT-Short '08, 101–104. Stroudsburg, PA, USA: Association for Computational Linguistics.

McIntosh, T., and Curran, J. R. 2008. Weighted mutual exclusion bootstrapping for domain independent lexicon and template acquisition. In *Proceedings of the Australasian Language Technology Association Workshop 2008*, 97–105.

McIntosh, T., and Curran, J. R. 2009. Reducing semantic drift with bagging and distributional similarity. 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 1 - Volume 1*, ACL '09, 396–404. Stroudsburg, PA, USA: Association for Computational Linguistics.

McIntosh, T. 2010. Unsupervised discovery of negative categories in lexicon bootstrapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, 356–365. Stroudsburg, PA, USA: Association for Computational Linguistics.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Commun. ACM* 38(11):39–41.

Murphy, T., and Curran, J. 2007. Experiments in mutual exclusion bootstrapping. Proceedings of the Australasian Language Technology Workshop 2007, 66–74.

Paşca, M. 2007. Weakly-supervised discovery of named entities using web search queries. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, 683–690. New York, NY, USA: ACM.

Pantel, P.; Crestan, E.; Borkovsky, A.; Popescu, A.-M.; and Vyas, V. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, 938–947. Stroudsburg, PA, USA: Association for Computational Linguistics.

Pasca, M. 2004. Acquisition of categorized named entities for web search. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, 137–145. New York, NY, USA: ACM.

Phillips, W., and Riloff, E. 2002. Exploiting strong syntactic heuristics and co-training to learn semantic lexicons. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, 125–132. Association for Computational Linguistics.

Qadir, A., and Riloff, E. 2012. Ensemble-based semantic lexicon induction for semantic tagging. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, 199–208. Montréal, Canada: Association for Computational Linguistics.

Riloff, E., and Jones, R. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, 474–479. Menlo Park, CA, USA: American Association for Artificial Intelligence.

Riloff, E., and Shepherd, J. 1997. A corpus-based approach for building semantic lexicons. In *Second Conference on Empirical Methods in Natural Language Processing*, 117–124.

Roark, B., and Charniak, E. 1998. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, 1110–1116. Stroudsburg, PA, USA: Association for Computational Linguistics.

Sarmento, L.; Jijkuon, V.; de Rijke, M.; and Oliveira, E. 2007. "more like these": Growing entity classes from seeds. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, 959–962. New York, NY, USA: ACM.

Thelen, M., and Riloff, E. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, 214–221. Stroudsburg, PA, USA: Association for Computational Linguistics.

Wang, R. C., and Cohen, W. W. 2007. Language-independent set expansion of named entities using the web. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, 342–350. Washington, DC, USA: IEEE Computer Society.