Minimally Supervised Classification to Semantic Categories using Automatically Acquired Symmetric Patterns

Roy Schwartz[1] **Roi Reichart**[2] **Ari Rappoport**[1]

[1]Institute of Computer Science, The Hebrew University
{roys02|arir}@cs.huji.ac.il

[2]Technion IIT
roiri@ie.technion.ac.il

Abstract

Classifying nouns into semantic categories (e.g., animals, food) is an important line of research in both cognitive science and natural language processing. We present a minimally supervised model for noun classification, which uses symmetric patterns (e.g., "X and Y") and an iterative variant of the k-Nearest Neighbors algorithm. Unlike most previous works, we do not use a predefined set of symmetric patterns, but extract them automatically from plain text, in an unsupervised manner. We experiment with four semantic categories and show that symmetric patterns constitute much better classification features compared to leading word embedding methods. We further demonstrate that our simple k-Nearest Neighbors algorithm outperforms two state-of-the-art label propagation alternatives for this task. In experiments, our model obtains 82%-94% accuracy using as few as four labeled examples per category, emphasizing the effectiveness of simple search and representation techniques for this task.

1 Introduction

The role of language is to express meaning. In the field of NLP, there has been an increasingly growing number of approaches that deal with semantics. Among these are vector space models (Turney and Pantel, 2010; Baroni and Lenci, 2010), lexical acquisition (Hearst, 1992; Dorow et al., 2005; Davidov and Rappoport, 2006), universal cognitive conceptual annotation (Abend and Rappoport, 2013) and automatic induction of feature representations (Collobert et al., 2011). In this paper, we utilize extremely weak supervision to classify words into fundamental cognitive semantic categories.

There are several types of semantic categories expressed by languages, e.g., objects, actions, and properties. We follow human development, acquiring coarse-grained categories and distinctions before detailed ones (Mandler, 2004). Specifically, we focus on the major class of concrete "*things*" (Langacker, 2008, Ch. 4), roughly corresponding to nouns – the main participants in linguistic clauses – that are universally present in the semantics of virtually all languages (Dixon, 2005).

Most works on noun classification to semantic categories require large amounts of human annotation to build training corpora for supervised algorithms (Bowman and Chopra, 2012; Moore et al., 2013) or rely on language-specific resources such as WordNet (Evans and Orăsan, 2000; Orăsan and Evans, 2007). Such heavy supervision is labor intensive and makes these models domain and language dependent.

Our reasoning is that weak supervision is highly valuable for semantic categorization, as it can compensate for the lack of input from the senses in text corpora. Our model therefore performs semantic category classification using only a small number of labeled seed words per category. The experiments we conduct show that such weak supervision is sufficient to construct a high quality classifier.

A key component of our model is the application of symmetric patterns. We define patterns to be sequences of words and wildcards (e.g., "X is a dog", "both X and Y", etc.). Accordingly, *symmetric* patterns are patterns that contain exactly two wildcards, where both wildcards are interchangeable. Examples of symmetric patterns include "X and Y", "X as well as Y" and "neither X nor Y".

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers,
pages 1612–1623, Dublin, Ireland, August 23-29 2014.

Works that apply symmetric patterns in their model generally require expert knowledge in the form of a pre-compiled set of patterns (Widdows and Dorow, 2002; Kozareva et al., 2008). In this work, we extract symmetric patterns in an unsupervised manner using the (Davidov and Rappoport, 2006) algorithm. This algorithm automatically extracts a set of symmetric patterns from plain text using simple statistics about high and low frequency word co-occurrences. The unsupervised nature of our approach makes it domain and language independent.

Our model addresses semantic classification in a transductive setup. It takes advantage of word similarity scores that are computed based on symmetric pattern features, and propagates information from concepts with known classes to the rest of the concepts. For this aim we apply an iterative variant of the k-Nearest Neighbors algorithm (denoted with I-k-NN) to a graph in which vertices correspond to nouns and word pairs are connected with edges based on their participation in symmetric patterns.

We experiment with a subset of 450 nouns from the CSLB dataset (Devereux et al., 2013), which were annotated with semantic categories by thirty human subjects. From the set of semantic categories in this dataset, we select categories that are both frequent and have a high inter-annotator agreement (Section 2). This results in a set of four semantic categories – *animacy, edibility, is_a_tool* and *is_worn*.

Our experiments show that our model performs very well even when only a small number of labeled seed words are available. For example, on the task of binary classification with respect to a single category, when using as few as four labeled seed words, classification accuracy reaches 82%-94%.

Furthermore, our model outperforms several strong baselines for this task. First, we compare our model against a model that uses a deep neural network word embedding baseline (Collobert et al., 2011) instead of our symmetric pattern based features, and applies the exact same I-k-NN algorithm. In recent years, deep networks word embeddings obtained state-of-the-art results in several NLP tasks (Collobert and Weston, 2008; Socher et al., 2013). However, in our task, features based on simple, intuitive and easy to compute symmetric patterns, lead to substantially better performance (average improvement of 0.15 F1 points). Second, our model outperforms two baseline models that utilize the same symmetric pattern classification features as in our model, but replace our simple I-k-NN algorithm with two leading label propagation alternatives (the normalized graph cut (N-Cut) algorithm (Yu and Shi, 2003) and the Modified Adsorption (MAD) algorithm (Talukdar and Crammer, 2009)). The average improvement over these two baselines is 0.21 and 0.03 F1 points .

The rest of the paper is organized as follows. Section 2 describes our semantic classification task and, particularly, the semantic classes that we aim to learn. Section 3 presents our method for automatic symmetric patterns acquisition. Sections 4, 5 and 6 describe our model, experimental setup and results, respectively. Related work is finally surveyed in Section 7.

2 Task Definition

The task we tackle in this paper is the classification of nouns into semantic categories. This section defines the categories we address and the dataset we use.

Semantic Categorization of Concrete Nouns. We focus on concrete "*things*" (Langacker, 2008), which correspond to *noun* categories. Nouns are interesting because they are the most basic lexical semantic categories. Specifically, children acquire nouns before any other category (Clark, 2009). Moreover, noun categories are generally not subjective. For example, it is hard to argue that a dog is not an animal, or that an apple is inedible, in most reasonable contexts. The context independent nature of nouns makes them appropriate for a type level classification task, such as the one we tackle. In order to provide a better description of the categories we aim to predict, we now turn to discuss the CSLB dataset, with which we experiment.

Dataset. We experiment with the CSLB property norms dataset (Devereux et al., 2013). In order to prepare this data set, thirty human subjects were presented with 638 concrete nouns and were asked to write the categories associated with each concept. Table 1 presents the top five categories for the nouns *apple* and *horse*.

Noun	Categories
Apple	is_a_fruit, does_grow_on_trees, is_green, is_red, has_pips_seeds
Horse	is_ridden, is_an_animal, has_four_legs, has_legs, has_hooves

Table 1: Five most frequent semantic categories for the words *apple* and *horse* in the CSLB dataset.

Category Selection. The CSLB dataset consists of a total of 2725 semantic categories. We apply a selection mechanism that provides us with a dataset in which (1) only noun categories (*things*) are included; and (2) only semantic categories that are prominent across humans are considered. For this, we apply the following filtering stages. First, since the vast majority of annotated categories are rare (for example, 1691 categories are assigned to a single noun only), we set a minimum threshold of 35 nouns per category (5% of the nouns). After removing highly infrequent categories, 28 are left. We then apply an inter-annotator agreement criterion: for each semantic category c, we compute the average number of human annotators that associated this category with a given noun, across the nouns annotated with c. We select the category c only if the value of this statistic is higher than 10 subjects ($1/3$ of the subjects), which results in a semantic category set of size 18. Finally, we discard categories, such as *color* and *size*, that do not correspond to *things*. We are left with four noun semantic categories: *animacy* (animals), *edibility* (food items), *is_a_tool* (tools), and *is_worn* (clothes).

Interestingly, the resulting semantic categories can also be justified from a cognitive perspective. There is a large body of work indicating that our categories relate to brain organization principles. For example, Just et al. (2010) showed that food products and tools arouse different brain activation patterns. Moreover, a number of works showed that both animate objects and tools are represented in specific brain regions. These works used neuroimaging methods such as functional magnetic resonance imaging (fMRI) (Naselaris et al., 2012), electroencephalography (EEG) (Chan et al., 2011) and magnetoencephalography (MEG) (Sudre et al., 2012). See (Martin, 2007) for a detailed survey. This parallel evidence to the prominence of our categories provides substance for intriguing future research.

3 Symmetric Patterns

Patterns. In this work, patterns are combinations of words and wildcards, which provide a structural phrase representation. Examples of patterns include "*X and Y*", "*X such as Y*", "*X is a country*", etc. Patterns can be used to extract various relations between words. For example, patterns such as "X of a Y" ("basement *of a* building") can be useful for detecting the meronymy (part-of) relation (Berland and Charniak, 1999). Symmetric patterns (e.g., "X *and* Y", "France *and* Holland"), which we use in this paper, can be used to detect semantic similarity between words (Widdows and Dorow, 2002).

Symmetric Patterns. *Symmetric* patterns are patterns that contain exactly two wildcards, and where these wildcards are interchangeable. Examples of symmetric patterns include "X *and* Y", "X *or* Y" and "X *as well as* Y". Previous works have shown that word pairs that participate in symmetric patterns bare strong semantic resemblance, and consequently, that these patterns can be used to cluster words into semantic categories, where a high precision, but low coverage (recall) solution is good enough (Dorow et al., 2005; Davidov and Rappoport, 2006). A key observation of this paper is that symmetric patterns can be also used for semantic classification, where recall is as important as precision.

Flexible Patterns. It has been shown in previous work (Davidov and Rappoport, 2006; Turney, 2008; Tsur et al., 2010; Schwartz et al., 2013) that patterns can be extracted from plain text in a fully unsupervised manner. The key idea that makes this procedure possible is the concept of "flexible patterns", which are composed of high frequency words (HFW) and content words (CW). Every word in the language is defined as either HFW or CW, based on the number of times this word appears in a large corpus. This clustering procedure is applied by traversing a large corpus, and marking words that appear with corpus frequency higher than a predefined threshold t_1 as HFWs, and words with corpus frequency lower than t_2 as CWs.[1]

[1] We follow (Davidov and Rappoport, 2006) and set $t_1 = 10^{-5}, t_2 = 10^{-3}$. Note that some words are marked both as HFW and as CW. See (Davidov and Rappoport, 2008) for discussion.

The resulting clusters have a desired property: HFWs are comprised mostly of function words (prepositions, determiners, etc.) while CWs are comprised mostly of content words (nouns, verbs, adjectives and adverbs). This coarse grained clustering is useful for pattern extraction from plain text, since language patterns tend to use fixed function words, while content words change from one instance of the pattern to another (Davidov and Rappoport, 2006).

Flexible patterns are extracted by traversing a large corpus and, based on the clustering of words to CWs and HFWs, extracting all pattern instances. An extracted pattern instance consists of CW wildcards and the actual words replacing the HFWs in the pattern type. Consider the sentence "*The boy is happy and joyful*". Replacing the content words with the CW wildcard results in "*The* CW *is* CW *and* CW". From this intermediate representation, we extract word sequences of a given length constraint and denote them as flexible patterns.[2] The flexible patterns of length 5 extracted from this sentence are "*The* CW *is* CW *and*" and "CW *is* CW *and* CW". The reader is referred to (Davidov and Rappoport, 2006) for more details.

Automatically Extracted Symmetric Patterns. Most models that incorporate symmetric patterns use a predefined set of patterns (Widdows and Dorow, 2002; Kozareva et al., 2008). In this work, we apply an automatic, completely unsupervised procedure for symmetric pattern extraction. This procedure, described in Algorithm 1, is adopted from (Davidov and Rappoport, 2006).

The procedure first extracts flexible patterns that contain exactly two CW wildcards. It then selects those flexible patterns in which both CWs are interchangeable. That is, it selects a pattern p if every word pair CW_1, CW_2 that participates in p indicates with high probability that the word pair C_2, C_1 also participates in p. For example, for the symmetric pattern "CW *and* CW", both "cats *and* dogs" and "dogs *and* cats" are semantically plausible expressions, and are therefore likely to appear in a large corpus. On the other hand, the flexible pattern "CW *such as* CW" is asymmetric, as exemplified in expressions like "countries *such as* France", where replacing the CWs does not result in a semantically plausible expression (# "France *such as* countries"). The selection process is done by computing the proportion of CW_1, CW_2 pairs that participate in p for which CW_2, CW_1 also participates in p. Patterns for which this proportion exceed a certain threshold are selected.

We apply Algorithm 1 on the google books 5-gram corpus (Michel et al., 2011)[3] and extract 20 symmetric patterns. Some of the more interesting symmetric patterns extracted using this algorithm include "CW *and the* CW", "*from* CW *to* CW", "CW *rather than* CW" and "CW *versus* CW". In the next section we present our approach to semantic classification, which makes use of automatically acquired symmetric patterns for word similarity computations.

4 Model

In this section we present our model for binary word classification according to a single semantic category in a minimally-supervised, transductive setup. Given a set of words, we label a small number of words with their correct label according to the category at hand (+1 for words that belong to the category, -1 for words that do not belong to it). Our model is based on an undirected weighted graph, in which vertices correspond to words, and edges correspond to relations between words. Our goal is to classify the unlabeled words (vertices) in the graph through a label propagation process. We now turn to describe our model in detail.

Graph Construction. We construct our graph such that an edge is added between two words (vertices) if both words participate in a symmetric pattern. The edge generation process is performed as follows. We first apply our symmetric pattern extraction procedure (Algorithm 1), and denote the set of selected symmetric patterns with P. We then traverse a large corpus[4] and extract all word pairs that participate in any pattern $p \in P$. We denote the number of occurrences of a word pair (w_1, w_2) in such patterns with f_{w_1,w_2}. Finally, we select all word pairs (w_1, w_2) for which $min(f_{w_1,w_2}, f_{w_2,w_1}) > \alpha$. Each such

[2] We set the maximal flexible pattern length to be 5.

[3] `https://books.google.com/ngrams`

[4] We use google books 5-grams (Michel et al., 2011).

Algorithm 1 Symmetric pattern extraction

```
procedure EXTRACT_SYMMETRIC_PATTERNS($C, W$)
    ▷ $C$ is a large corpus, $W$ is a lexicon
    ▷ Traverse $C$ and extract all flexible patterns of length 3-5 that appear in $C$ and contain exactly two content words
    $P \leftarrow$ extract_flexible_patterns($C, W$)
    for $p \in P$ do
        if $p$ appears in $<10^{-6}$ of the sentences in $C$ then
            Discard $p$ and continue
        end if
        $G_p \leftarrow$ a directed graph s.t. $V(G_p) \leftarrow W, E(G_p) \leftarrow \{(w_1, w_2) \in W^2 : w_1, w_2$ participate in at least one instance of $p\}$
        ▷ An undirected graph based on the bidirectional edges of the $G_p$
        $symG_p \leftarrow$ an undirected graph: $\{(w_1), (w_1, w_2) : (w_1, w_2) \in E(G_p) \wedge (w_2, w_1) \in E(G_p)\}$
        ▷ Two measures of symmetry
        $M_1 \leftarrow \frac{|V(symG_p)|}{|V(G_p)|}, M_2 \leftarrow \frac{|E(symG_p)|}{|E(G_p)|}$
        ▷ Symmetric pattern candidates are those with high $M_1$ and $M_2$ values
        if $\min(M_1, M_2) < 0.05$ then
            Discard $p$
        end if
    end for
    for $p \in P$ do
        ▷ E.g., "CW and CW" is contained in "both CW and CW"
        if $\exists p' \in P$ s.t. $p'$ is contained in $p$ then
            Discard $p$
        end if
    end for
    return The top 20 members of $P$ with the highest $M_1$ value
end procedure
```

pair is connected with an edge e_{w_1,w_2} in the graph, where the edge weight (denoted with w_{w_1,w_2}) is the geometric mean between f_{w_1,w_2} and f_{w_2,w_1}.

Label Propagation. Given a small number of annotated words (vertices), our goal is to propagate the information these words convey to other words in the graph. To do so, we apply an iterative variant of the k-Nearest Neighbors algorithm (I-k-NN). This iterative variant is required due to graph sparsity; when starting with a small set of labeled vertices, most unlabeled vertices do not have any labeled neighbor, and thus running the standard k-NN algorithm would result in classifying a very small number of vertices. Our approach is to run iterations of the k-NN algorithm, and thus propagate information to additional vertices at each iteration. At each k-NN step, the algorithm selects words that have at least one labeled neighbor. From this set, only the words that have the highest ratio of neighbors with the same label are selected, and are assigned with this label.

Consider a simple example. Say we have three candidate vertices a, b and c, where a has one neighbor with label +1 ($ratio(a) = 1/1 = 1.0$), b has two neighbors with label -1 ($ratio(b) = 2/2 = 1.0$) and c has three neighbors with label +1 and one neighbor with label -1 ($ratio(c) = max(3, 1)/4 = 3/4$). Then, a and b are selected and are assigned with $+1$ and -1, respectively.

Seed Expansion. In minimally supervised setups like ours, the model is initialized with a small set of labeled seed examples. A natural approach in such settings is to apply a seed expansion step, in order to obtain a larger set of labeled seeds. Our method uses the same graph construction procedure described above, but uses a larger edge generation threshold $\beta >> \alpha$.[5] We then apply an iterative procedure that labels a vertex v with a label l if either (a) v is directly connected to γ of the vertices labeled with l or (b) v is connected to δ_l of the neighbors of vertices labeled with l.[6] This procedure is run iteratively until no more vertices meet any of the criteria (a) or (b).

[5] Using a larger threshold results in a sparser graph. Nevertheless, each edge in this graph is more likely to represent a real semantic relation.

[6] γ and δ_l are hyperparameters tuned on our development set (see Section 5.2).

5 Experimental Setup

5.1 Baselines

We compare our model to two types of baselines. The first (Classification Features Baselines) utilizes the I-k-NN algorithm, along with a different set of classification features. The second (Label Propagation Baselines) utilizes the same classification features as we do, but replaces I-k-NN with a more sophisticated label propagation algorithm.

5.1.1 Classification Features Baselines

In this set of baselines, we use different methods for building our graph. Concretely, instead of adding edges for pairs of words that appear in the same symmetric pattern, we use word similarity measures based on different feature sets as described below. The process of building the graph using the baseline word similarity measures is described in Section 5.2.

SENNA. Deep neural networks have gained recognition as leading feature extraction methods for word representation (Collobert and Weston, 2008; Socher et al., 2013). We use SENNA,[7] a deep network based word embedding method, which has been used to produce state-of-the-art results in several NLP tasks, including POS tagging, chunking, NER, parsing and SRL (Collobert et al., 2011). We use the cosine similarity between two word embeddings as a word similarity measure.

Brown. This baseline is derived from the clustering induced by the Brown algorithm (Brown et al., 1992).[8] This clustering, in which words share a cluster if they tend to appear in the same lexical context, has shown useful for several NLP tasks, including POS tagging (Clark, 2000), NER (Miller et al., 2004) and dependency parsing (Koo et al., 2008). We use it in order to control for the possibility that a simple contextual preference similarity correlates with similarity in semantic categorization better than symmetric pattern features.

The Brown algorithm builds a binary tree, where words are located at leaf nodes. We use the graph distance between two words u, v (i.e., the shortest path length between u, v in the tree) as a word similarity measure for building our graph.

5.1.2 Label Propagation Baselines

In this type of baselines, we replace I-k-NN with a different label propagation algorithm, while still using the symmetric pattern features for word similarity computations.

N-Cut. This baseline applies the normalized graph cut algorithm (Yu and Shi, 2003)[9] for label propagation. Given a graph $G = (V, E)$ and two sets of vertices $A, B \subseteq V$, this algorithm defines $links(A, B)$ to be the sum of edge weights between A and B. The objective of the algorithm is to find the clusters $A, V \setminus A$ that minimize $\frac{links(A,V \setminus A)}{links(A,V)}$. The algorithm of (Yu and Shi, 2003) is particularly efficient for this problem as it avoids eigenvector computations which may become computationally prohibitive for large graphs (for more details, see their paper). In order to encode information about our labeled seed words, we hard-code a large negative value (-100000) to the weights of edges between seed words with different labels (positive and negative).

MAD. The Modified Adsorption (MAD) algorithm (Talukdar and Crammer, 2009)[10] is an extension of the Adsorption algorithm (Baluja et al., 2008). MAD is a stochastic graph-based label propagation algorithm which has shown to have a number of attractive theoretical properties and demonstrated good experimental results.

[7] The word embeddings were downloaded from `http://ml.nec-labs.com/senna/`

[8] We use the clusters induced by (Koo et al., 2008), who applied the Brown algorithm implementation of (Liang, 2005) to the BLLIP corpus (Charniak et al., 2000). `http://www.people.csail.mit.edu/maestro/papers/bllip-clusters.gz`

[9] `http://www.cis.upenn.edu/~jshi/software/Ncut_9.zip`

[10] `http://github.com/parthatalukdar/junto`

5.2 Experiments

Graph Construction. We experiment with the CSLB dataset (Devereux et al., 2013), consisting of 638 nouns, annotated with their semantic categories by thirty human subjects. We first omit all nouns that are annotated as having more than one sense, and use the remaining 603 nouns to build our graph. From these nouns, 146 nouns are annotated as animate, 115 as edible, 50 as wearable and 35 as tools.[11] We then discard nouns that have less than two neighbors, which results in a final set of 450 nouns (vertices).

The graphs used in the classification features baselines are different than those used by the models that use our symmetric pattern classification features, since the features define the graph structure (Section 4). In order to provide a meaningful comparison, we build graphs with the same number of vertices for each of these baselines. We do so by selecting the n edges with the highest weight, together with the set of vertices connected by these edges, such that the resulting graph has 450 vertices. Working with these sets of vertices is the optimal setting for these baselines, as the resulting graphs are the ones with the highest possible edge weights for graphs with 450 vertices.[12]

Parameter Tuning. In order to avoid adding additional labeled examples for the sake of parameter tuning, we set the hyperparameter values to the ones for which each model performs best on an auxiliary semantic classification task. Concretely, we experiment with a fifth semantic category (*audibility*),[13] which is not part of our evaluation setting, for parameter tuning. Note that this results in our model having the same hyperparamter values for all four classification tasks.

In order to ensure that the models assign all participating words with labels, we set α=3, where α is the minimal number of times a word pair should appear in the same symmetric pattern in order to have an edge in our graph (See Section 4). In our seed expansion procedure, where we search for seeds whose label is predicted with high confidence, only word pairs that appear at least β=50 times in the same symmetric pattern are assigned an edge in the graph. We set the seed expansion procedure parameters to be $\gamma = 0.6, \delta_{+1} = 0.5, \delta_{-1} = 0.2$.

Evaluation. For each classification task, we run experiments with 4, 10, 20 and 40 labeled seed words. In each setting, half of the labeled seed words are assigned a positive label and the other half are assigned a negative label. For each semantic category and labeled seed set size, we repeat our experiment 1000 times, each of which with a different set of randomly selected labeled seed examples, and report the average results. We report both accuracy (number of correct labels divided by number of vertices in the graph) and F1 score, which is the harmonic mean of p (the average precision across labels) and r (average recall across labels).

These two measures represent complementary aspects of our results. On the one hand, accuracy is the most natural classification performance measure. On the other hand, the number of positive labels is substantially smaller than the number of negative labels,[14] and thus this measure can be manipulated: a dummy model that always assigns the negative label gets a high accuracy. The F1 score controls against such models by assigning them low scores.

6 Results

Our experiments are designed to explore two main questions: (a) the value of symmetric patterns as semantic classification features, compared to state-of-the-art word clustering and embedding methods; and (b) the required complexity of an algorithm that can propagate information about semantic similarity. Particularly, we test the value of our simple I-k-NN algorithm compared to more sophisticated alternatives.

A Minimally Supervised Setting. Our first set of experiments is in a minimally supervised setting where only two positive and two negative examples are available for each binary classification task. This

[11] Some words are classified as belonging to more than one category (e.g., "chicken" is both animate and edible).

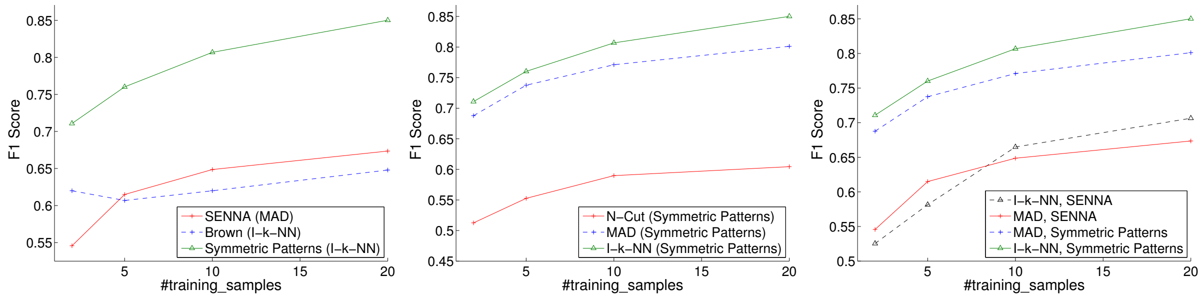
[12] The resulting graphs are actually denser than the symmetric patterns-based graph: 14K and 9K edges for the Brown and SENNA graphs, respectively, compared to $<$ 5K edges in the symmetric patterns graph.

[13] We used four labeled seed words in these experiments.

[14] Only 6-25% of the nouns have a positive label.

		Animacy			Edibility			is_worn			is_a_tool		
		SP	SENNA	Brown	SP	SENNA	Brown	SP	SENNA	Brown	SP	SENNA	Brown
Acc.	MAD	80.4%	77.7%	12.0%	75.0%	56.5%	14.8%	82.7%	66.8%	14.7%	73.3%	67.7%	12.2%
	N-Cut	71.4%	60.4%	51.2%	75.5%	59.4%	50.9%	83.3%	71.5%	51.4%	**82.7%**	77.1%	52.0%
	I-k-NN	**85.1%**	76.0%	55.5%	**82.2%**	56.8%	68.0%	**94.1%**	70.9%	66.7%	82.0%	75.7%	65.0%
F1	MAD	0.77	0.76	0.18	0.69	0.55	0.24	0.71	0.56	0.22	0.58	0.47	0.17
	N-Cut	0.49	0.45	0.46	0.51	0.44	0.45	0.61	0.56	0.41	0.56	0.50	0.38
	I-k-NN	**0.78**	0.70	0.48	**0.71**	0.53	0.62	**0.86**	0.59	0.55	**0.64**	0.52	0.51

Table 2: Accuracy and F1 score comparison between our model and the baselines. The columns correspond to the type of classification features used by the model: SP – symmetric patterns, SENNA – word embeddings extracted using deep networks (Collobert et al., 2011), Brown – Brown word clustering (Brown et al., 1992). The rows correspond to the algorithms applied by the model: N-Cut – the normalized graph cut algorithm (Yu and Shi, 2003), MAD – the modified adsorption algorithm (Talukdar and Crammer, 2009), I-k-NN – our iterative k-NN algorithm. Our model (I-k-NN + SP) is superior in all cases, except for the accuracy of the "is_a_tool" semantic category, where it is second only to N-Cut+SP.



(a) Classification Features Comparison (b) Algorithm Comparison (c) Top four Best Models

Figure 1: (a) Comparison of the different classification features. The figure shows the F1 scores of the best model that uses each of the feature sets (the label propagation algorithm used in each model appears in parentheses). (b) Comparison of the different label propagation algorithms. The figure shows the F1 scores of the best model that uses each of the algorithms (the classification feature sets used in each model appears in parentheses. It is always symmetric patterns). (c) The four best overall models (algorithm + classification feature set). The figures show that the symmetric pattern feature set is superior to the other feature sets, and that I-k-NN is superior or comparable to the other label propagation algorithms.

setup enables us to explore the performance of our model when the amounts of labeled training data is taken to the possible minimum.

Table 2 presents our results. With respect to objective (a), the table clearly demonstrates that symmetric patterns lead to much better results compared to the alternatives. Particularly, for all four semantic categories, and across both evaluation measures, it is a model that utilizes symmetric pattern classification features that achieves the best results. The average difference between the best model that uses symmetric patterns and the best model that does not is 12.5% accuracy and 0.13 F1 points. The dominance of symmetric pattern classification features is further demonstrated by the fact that a model that uses these features always performs better than a model that uses the same algorithm but different features.

With respect to objective (b) the table shows that I-k-NN provides a large improvement in seven out of eight (*category* × *evaluation measure*) settings. The average difference between the best model that utilizes I-k-NN and the best model that applies a different algorithm is 5.4% accuracy and 0.06 F1 points.

Analysis of Labeled Seed Set Size. In order to get a wider perspective on our model, we repeated our experiments with various sizes of the labeled seed set: 5,10 and 20 positive and negative labeled examples per semantic category. For brevity, only the F1 score results of the edibility category are presented. The trends observed on the other semantic categories (as well as when using the accuracy measure) are very similar.

Figure 1a compares the different classification features. For each feature f, results of the best performing model that uses f are shown. The results reveal that symmetric patterns clearly outperform the other features. The average differences between the best symmetric patterns-based model and the best models that use the other features are 0.15 (SENNA) and 0.16 (Brown) F1 points.

Figure 1b compares the different label propagation algorithms. For each algorithm a, results for the best performing model that uses a are presented. The results reveal that the I-k-NN algorithm outperforms both algorithms by 0.03 (MAD) and 0.21 (N-Cut) F1 points. The results also show that for all algorithms, the best performing model uses symmetric patterns classification features, which further demonstrates the dominance of these features.

Finally, Figure 1c presents the four top performing models (algorithm + classification feature). In accordance with the other findings presented in this section, the top two models, which outperform the other models by a large margin, apply symmetric pattern classification features.

Seed Expansion Effect. Our model uses a seed expansion procedure in order to expand a small set of labeled seed words to a larger set (see Section 4). In order to assess the quality of this procedure we compute, for each semantic category, the average size of the expanded set and the accuracy of the new seeds (i.e., the proportion of new seeds that are labeled correctly). Results show that the initial set is increased from four seeds (two positive + two negative) to 48-52, and that the accuracy of the new seeds is as high as 88-99%. Our experiments also show that this procedure provides a substantial performance boost to our I-k-NN algorithm, which obtains a 7.2% accuracy and 0.05 F1 points improvement (averaged over the four semantic categories) when applied with the expanded set of labeled seed words compared to the original set of size four.

7 Related Work

Classification into Semantic Categories. Several works tackled the task of semantic classification, mostly focusing on animacy, concreteness and countability. The vast majority of these works are either supervised (Hatzivassiloglou and McKeown, 1997; Baldwin and Bond, 2003; Peng and Araki, 2005; Øvrelid, 2005; Nagata et al., 2006; Xing et al., 2010; Kwong, 2011; Bowman and Chopra, 2012) or make use of external, language-specific resources such as WordNet (Orăsan and Evans, 2001; Orăsan and Evans, 2007; Moore et al., 2013). Our work, in contrast, is minimally supervised, requiring only a small set of labeled seed words.

Ji and Lin (2009) classified words into the gender and animacy categories, based on their occurrences in instances of hand-crafted patterns such as "X *who* Y" and "X *and his* Y". While their model uses patterns that are tailored to the animacy and gender categories, our model uses automatically induced patterns and is thus applicable to a range of semantic categories.

Finally, Turney et al. (2011) built a label propagation model that utilizes LSA (Landauer and Dumais, 1997) based classification features. They used their model to classify nouns into the concrete/abstract category using 40 labeled seed words . Unlike our model, which requires only a small set of labeled seeds, their algorithm is actually heavily supervised, requiring thousands of labeled examples for selecting the seed set of labeled words that are used for propagation. Our model, on the other hand, does not require any seed selection procedure, and utilizes a randomly selected set of labeled seed words.

Lexical Acquisition. Another line of work focused on the acquisition of semantic categories. In this setup, a model aims to find a core seed of words belonging to a given category, sacrificing recall for precision. Our model tackles a different task, namely the classification of words according to a given category where both recall and precision are to be optimized.

Lexical acquisition models are either supervised (Snow et al., 2006), unsupervised, making use of symmetric patterns (Davidov and Rappoport, 2006), or lightly supervised, requiring expert, language specific knowledge for compiling a set of hand-crafted patterns (Widdows and Dorow, 2002; Kozareva et al., 2008; Wang and Cohen, 2009). Other models require syntactic annotation derived from a supervised parser to extract coordination phrases (Riloff and Shepherd, 1997; Dorow et al., 2005). Our model automatically induces symmetric patterns, obtaining high quality results without relying on any type of language specific knowledge or annotation. Moreover, some of the works mentioned above (Riloff and Shepherd, 1997; Widdows and Dorow, 2002; Kozareva et al., 2008) also require manually selected label seeds to achieve good performance; in contrast, our work performs very well with a randomly selected set of labeled seed words.

8 Conclusion

We presented a minimally supervised model for noun classification into coarse grained semantic categories. Our model obtains 82%-94% accuracy on four semantic categories even when using only four labeled seed words per category. We showed that our modeling decisions – using symmetric patterns as classification features and a simple iterative k-NN algorithm for label propagation – lead to a substantial performance gain compared to state-of-the-art, more sophisticated, alternatives. Our results demonstrate the applicability of minimally supervised methods for semantic classification tasks. Future work will include modifying our model to support other, more fine-grained types of semantic categories, including adjectival categories (*properties*). We also plan to work on token-level word classification, and thus support multi-sense words, as well as demonstrate the power of unsupervised patterns acquisition for multilingual setups.

Acknowledgments

This research was funded (in part) by the Harry and Sylvia Hoffman leadership and responsibility program (for the first author), the Google Faculty research award (for the second author), the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI) and the Israel Ministry of Science and Technology Center of Knowledge in Machine Learning and Artificial Intelligence (Grant number 3-9243).

References

O. Abend and A. Rappoport. 2013. Universal conceptual cognitive annotation (UCCA). In *Proc. of ACL*.

T. Baldwin and F. Bond. 2003. A plethora of methods for learning English countability. In *Proc. of EMNLP*.

S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly. 2008. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proc. of WWW*, pages 895–904. ACM.

M. Baroni and A. Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

M. Berland and E. Charniak. 1999. Finding parts in very large corpora. In *Proc. of ACL*.

S. R. Bowman and H. Chopra. 2012. Automatic Animacy Classification. In *Proc. of NAACL-HLT Student Research Workshop*.

P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

A. M. Chan, J. M. Baker, E. Eskandar, D. Schomer, I. Ulbert, K. Marinkovic, S. S. Cash, and E. Halgren. 2011. First-pass selectivity for semantic categories in human anteroventral temporal lobe. *The Journal of Neuroscience*, 31(49):18119–18129.

E. Charniak, D. Blaheta, N. Ge, K. Hall, J. Hale, and M. Johnson. 2000. BLLIP 198789 WSJ Corpus Release 1, LDC No. LDC2000T43. Linguistic Data Consortium.

A. Clark. 2000. Inducing syntactic categories by context distribution clustering. In *Proc. of CoNLL*.

E. V. Clark. 2009. *First language acquisition*. Cambridge University Press.

R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of ICML*.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537.

D. Davidov and A. Rappoport. 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *Proc. of ACL-Coling*.

D. Davidov and A. Rappoport. 2008. Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated SAT analogy questions. In *Proc. of ACL-HLT*.

B. J. Devereux, L. K. Tyler, J. Geertzen, and B. Randall. 2013. The centre for speech, language and the brain (CSLB) concept property norms. *Behavior research methods*, pages 1–9.

R. M. Dixon. 2005. *A semantic approach to English grammar*. Oxford University Press.

B. Dorow, D. Widdows, K. Ling, J. P. Eckmann, D. Sergi, and E. Moses. 2005. Using Curvature and Markov Clustering in Graphs for Lexical Acquisition and Word Sense Discrimination.

R. Evans and C. Orăsan. 2000. Improving anaphora resolution by identifying animate entities in texts. In *Proc. of DAARC*.

V. Hatzivassiloglou and K. R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proc. of ACL*.

M. A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of Coling – Volume 2*.

H. Ji and D. Lin. 2009. Gender and Animacy Knowledge Discovery from Web-Scale N-Grams for Unsupervised Person Mention Detection. In *Proc. of PACLIC*.

M. A. Just, V. L. Cherkassky, S. Aryal, and T. M. Mitchell. 2010. A neurosemantic theory of concrete noun representation based on the underlying brain codes. *PloS one*, 5(1):e8622.

T. Koo, X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *Proc. of ACL-HLT*.

Z. Kozareva, E. Riloff, and E. Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proc. of ACL-HLT*.

O. Y. Kwong. 2011. Measuring concept concreteness from the lexicographic perspective. In *Proc. of PACLIC*.

T. K. Landauer and S. T. Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

R. W. Langacker. 2008. *Cognitive grammar: A basic introduction*. Oxford University Press.

P. Liang. 2005. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

J. M. Mandler. 2004. *The foundations of mind: Origins of conceptual thought*. Oxford University Press New York.

A. Martin. 2007. The representation of object concepts in the brain. *Annual Review of Psychology*, 58:25–45.

J. B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, et al. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.

S. Miller, J. Guinness, and A. Zamanian. 2004. Name tagging with word clusters and discriminative training. In *Proc. of NAACL*.

J. L. Moore, C. J. Burges, E. Renshaw, and W.-t. Yih. 2013. Animacy Detection with Voting Models. In *Proc. of EMNLP*.

R. Nagata, A. Kawai, K. Morihiro, and N. Isu. 2006. Reinforcing English countability prediction with one countability per discourse property. *Proc. of ACL-Coling*.

T. Naselaris, D. E. Stansbury, and J. L. Gallant. 2012. Cortical representation of animate and inanimate objects in complex natural scenes. *Journal of Physiology-Paris*, 106(5):239–249.

C. Orăsan and R. Evans. 2001. Learning to identify animate references. In *Proc. of the Workshop on Computational Natural Language*.

C. Orăsan and R. Evans. 2007. NP Animacy Identification for Anaphora Resolution. *JAIR*, 29:79–103.

L. Øvrelid. 2005. Animacy classification based on morphosyntactic corpus frequencies: some experiments with Norwegian nouns. In *Proc. of the Workshop on Exploring Syntactically Annotated Corpora*, pages 1–11.

J. Peng and K. Araki. 2005. Detecting the countability of english compound nouns using web-based models. In *Proc. of IJCNLP*.

E. Riloff and J. Shepherd. 1997. A corpus-based approach for building semantic lexicons. In *Proc. of EMNLP*.

R. Schwartz, O. Tsur, A. Rappoport, and M. Koppel. 2013. Authorship Attribution of Micro-Messages. In *Proc. of EMNLP*.

R. Snow, D. Jurafsky, and A. Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proc. of ACL-Coling*.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of EMNLP*.

G. Sudre, D. Pomerleau, M. Palatucci, L. Wehbe, A. Fyshe, R. Salmelin, and T. Mitchell. 2012. Tracking neural coding of perceptual and semantic features of concrete nouns. *NeuroImage*, 62(1):451–463.

P. P. Talukdar and K. Crammer. 2009. New regularized algorithms for transductive learning. In *ECML-PKDD*, pages 442–457. Springer.

O. Tsur, D. Davidov, and A. Rappoport. 2010. ICWSM – a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Proc. of ICWSM*.

P. D. Turney and P. Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

P. Turney, Y. Neuman, D. Assaf, and Y. Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proc. of EMNLP*.

P. D. Turney. 2008. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33:615–655.

R. C. Wang and W. W. Cohen. 2009. Automatic set instance extraction using the web. In *Proc. of ACL-IJCNLP*.

D. Widdows and B. Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proc. of Coling*.

X. Xing, Y. Zhang, and M. Han. 2010. Query difficulty prediction for contextual image retrieval. In *Advances in Information Retrieval*, pages 581–585. Springer.

S. X. Yu and J. Shi. 2003. Multiclass spectral clustering. In *Proc. of ICCV*.