

Automatic classification of patterns from the Pattern Dictionary of English Verbs

Ismail El Maarouf

University of Wolverhampton
i.el-maarouf@wlv.ac.uk

Vít Baisa

Masaryk University
Lexical Computing Ltd
xbaisa@fi.muni.cz

Abstract

The paper presents a supervised approach to semantic parsing, based on a new semantic resource, the Pattern Dictionary of English Verbs (PDEV). PDEV lists the most frequent patterns of English verbs identified in corpus. Each argument in a pattern is semantically categorized with semantic types from the PDEV ontology. Each pattern is linked to a set of sentences from the British National Corpus.

The article describes PDEV in details and presents the task of pattern classification. The system described is based on a distributional approach, and achieves 66% in Micro-average F1 across a sample of 25 of the most frequent verbs.

1 Introduction

This paper reports the results of Natural Language Processing (NLP) experiments in semantic parsing, based on a new semantic resource, the Pattern Dictionary of English Verbs (PDEV) (Hanks, 2013). This resource is the output of Corpus Pattern Analysis (CPA; (Hanks, 2004)), a corpus lexicography technique for mapping meaning onto words in text. CPA analyses the prototypical syntagmatic patterns with which words in use are associated. The patterns emerge from the analysis of corpus concordance lines and careful attention to linguistic context clues is applied to characterize pattern elements and to distinguish between patterns. Only in a second step is an “implicature” (i.e. a meaning) mapped onto a pattern. In other words, CPA is driven by syntagmatic patterns, not meaning.

Given these two features (pattern-driven and corpus-driven), this resource is unique in its kind, across languages. However, while CPA has made contributions to lexicography and to linguistics, no

experiments have yet been made in NLP to use PDEV in applications such as Information Extraction or Statistical Machine Translation.

The present paper proposes to make use of PDEV as a resource for the semantic processing of text. It describes its structure in detail (section 2) and proposes the task of Pattern classification as a first step in semantic parsing (section 3). Contributions are summarized in section 4.

2 Background

2.1 Corpus Pattern Analysis

PDEV is built using the CPA methodology, which draws on Corpus Linguistics (Sinclair, 1991), and is inspired by semantic theories such as the Generative Lexicon (Pustejovsky, 1995), Frame Semantics (Fillmore, 1985) and Preference Semantics (Wilks, 1975). As a methodology for building lexical resources, CPA takes the position that words are only meaningful in context: words in isolation tend to be ambiguous while word patterns are rarely so. While this may seem self-evident, it has important implications for lexical semantics, which are developed in the Theory of Norms and Exploitations (TNE) (Hanks, 2013). According to TNE, it is a fallacy to attempt the definition of words independently and outside of context. Words should be described according to, and along with, the patterns in which they are found in real language use.

CPA builds typical phraseological patterns from corpora, by clustering corpus tokens (labelling them) according to the similarity of their context. The similarity is evaluated in different steps.

- Syntactic analysis involves the identification of the main structures such as idiomatic expressions, phrasal uses, transitive/intransitive patterns, causative/inchoative alternations, and argument/adjunct discrimination.

nb	%	Pattern & primary implicature
1	7%	[[Plant]] blossom [NOOBJ] [[Plant]] produces flowers or masses of flowers
2	87%	[[Eventuality Psych]] blossom [NOOBJ] (into [[Anything = Good]]) [[Eventuality Psych]] develops in a promising way or into something that is expected or desired

[[Psych]] refers to Psychological Entities and includes Emotions, Attitude and Goal.

Figure 1: Patterns for the verb *blossom*

- Semantic analysis involves the use of semantic features shared by collocates in each argument position. For example, Semantic Types (ST; e.g. [[Human]], [[Building]], [[Event]]) are used to represent the prototypical properties shared by the collocates found in a specific pattern position.

Since PDEV patterns represent abstractly several features of tokens from a large sample, they are rarely fully instantiated in an example: actual examples most often instantiate part of a pattern (e.g. subject ellipsis).

2.2 The structure of patterns

PDEV is created using three main tools: a corpus interface, i.e. The SketchEngine¹ (Kilgarriff et al., 2004), an ontology of semantic types², and the pattern dictionary³. PDEV lexicographers use the British National Corpus⁴ (BNC), a large reference corpus containing various text types in British English (100 million words).

A verb pattern includes arguments such as Subject and Object. Each argument can be described according to determiners, semantic types, contextual roles, and lexical sets:

- Determiners account for distinctions between “take place” and “take **his** place”.
- Semantic types account for distinctions such as “building [[Machine]]” and “building [[Relationship]]”.
- Contextual roles account for distinctions such as “[[Human=Film Director]] shoot” and “[[Human=Sports Player]] shoot”.
- Lexical sets account for distinctions such as “reap the **whirlwind**” and “reap the **harvest**”.

¹<https://the.sketchengine.co.uk>

²http://nlp.fi.muni.cz/projekty/cpa/public_onto.html

³<http://deb.fi.muni.cz/pdev>

⁴<http://www.natcorp.ox.ac.uk/>

Figure 1 shows an example of the PDEV entry of the verb *to blossom*. Both patterns are intransitive; the first has the semantic type [[Plant]] as subject and may be classified as the literal meaning even though it is comparatively rare. The criterion that distinguishes the second pattern, which may be classified as a conventional metaphor, is the semantic type ([[Eventuality]] or [[Psych]]). Pattern 2 also includes an optional prepositional phrase as additional context clue.

- (1) The Times noted **fruit trees** which had begun to *blossom* ...
- (2) The **silk trade** *blossomed* in Blockley...

Pattern 2 (example 2) illustrates an alternation of semantic types. It means that in the whole set of lines tagged as “blossom 2”, subjects are either [[Eventuality]] or [[Psych]]. A semantic type provides the relevant semantic value of all words in context. They are, in practice, controlled generalizations of corpus observations.

Each pattern is described with (i) a primary implicature which elaborates the meaning of the pattern and (ii) percentages. Percentages are obtained by dividing the number of tagged lines over a random sample (the size of the sample depends on the frequency of a verb, usually 250 corpus lines).

3 Pattern classification

3.1 Description of the experiment

An important task performed by semantic parsers is Word Sense Disambiguation (WSD), in which systems predict the senses of words in text. WSD experiments (Navigli, 2009) have used WordNet⁵ as a sense repository but we decided to explore how PDEV patterns could be used in this context.

As each pattern is linked to a set of lines, the present task of *pattern classification* requires systems to identify the correct pattern for each verb token. Our experiment was carried out on 25 verbs

⁵<http://wordnet.princeton.edu/>

$$\text{Macro-F1} = \frac{1}{|C|} \cdot \sum_{k=1}^{|C|} \frac{(1 + \beta) \cdot \text{Precision}_k \cdot \text{Recall}_k}{\beta \cdot (\text{Precision}_k + \text{Recall}_k)}$$

with $\beta = 1$ (1)

with comparatively high frequency in the BNC, on a range of patterns. The dataset contains 20418 verb tokens and was split using the following stratified sampling method: tokens were randomly selected from each verb pattern separately, using a 0.8:0.2 ratio, making sure that in extreme cases, where the set included less than 4 instances, the training set would always contain at least as many examples as in the test set.

Two evaluation metrics were used: Micro-average (Micro-F1) and Macro-average F-score (Macro-F1). Micro-F1 can be computed by counting False and True positives and negatives across classes. Micro-F1 can be complemented with Mac-F1 which gives an estimate of the performance of systems in discriminating patterns (by giving equal weight to classes rather than to instances; see equation (1)).

The baseline was generated by applying the majority class (most frequent) found in the training set, to the test set. Since the dataset is highly biased in terms of label frequency, the baseline Micro-F1 is quite high (0.62 across verbs). However, the baseline reaches 0.12 in Macro-F1.

3.2 Bootstrapping system

The system used for this task is a solution available in the Sketch Engine (Kilgarriff and Rychly, 2010), a corpus query system allowing users to explore corpus concordances. This system bootstraps from an existing automatic thesaurus (Grefenstette, 1994; Lin, 1998) to assign a label to a given verb token. The thesaurus is based on a regular grammar which identifies collocates linked to a verb through a syntactic relation (such as *subject*). The system applies the grammar to extract dependency triples of the form $||w, r, w'||$, where w is a lemma linked to another lemma w' by a relation r . Each triple is weighted with an association score based on the set of extracted triples, as described in equation (2). A distance measure described in equation (3) is then applied to words sharing similar contexts (Rychly and Kilgarriff, 2007).

The bootstrapping algorithm uses the thesaurus scores to predict a label for each token. For each verb token v of the test set, it compares its contexts (r, w') to the contexts (r, w) labelled as k in the training set (of frequency over 1). The score for each token, results from the sum of the contexts having the best score as described in equation (4).

This method relies on the hypothesis that tokens sharing identical context should be labelled identically. It therefore does not normally discriminate cases where the same context is tagged with two different labels. This occurs only rarely. Two thresholds have been tested, `minscore`, the minimal score returned by the algorithm, and `mindiff`, the minimal difference between the best score and the second best score.

3.3 Results

The bootstrapping system was tested on several combinations of parameters `mindiff` and `minscore`. The best combination was `mindiff = 0.1`, thus a low difference between the first two scores returned by the algorithm, and `minscore = 0.9`, thus a high score threshold.

Table 1 shows that, on average, the system beats the baseline on both Micro-F1 and Macro-F1. While the difference (`diff`) between the system and the baseline in Micro-F1 is low, it is much higher for Macro-F1, which shows that the bootstrapping system is not biased towards the majority class.

Detailed analysis revealed that the bootstrapping generally suffers from fairly low recall, but has a very satisfying precision on average (Micro-Prec/Micro-Rec = 0.86/0.56; Macro-Prec/Macro-Rec = 0.56/0.41).

Conclusion

This article has presented new results for the classification of verb patterns from the Pattern Dictionary of English Verbs (PDEV). The latter is an interesting resource for semantic parsing as it is a corpus-based meaning repository with links to patterns of use. The tagged corpus of the PDEV has been used on a task of pattern classification similar to Word Sense Disambiguation, which is potentially beneficial to many Natural Language Processing applications.

The system used in this experiment is a bootstrapping algorithm relying on a distributional thesaurus and is a solution available in the

Verb	nb of Pat	Test size	Micro-Average F1			Macro-Average F1		
			System	Baseline	Diff	System	Baseline	Diff
blow	43	194	59	21	+38	40	10	+30
break	37	211	64	11	+53	41	3	+38
smile	29	101	37	79	-42	27	7	+20
laugh	18	160	45	65	-20	45	7	+38
sleep	16	168	62	79	-17	49	6	+43
object	14	123	59	88	-29	61	15	+46
breathe	12	203	62	40	+22	46	32	+14
arouse	11	102	90	94	-4	53	31	+22
beg	10	208	67	31	+36	52	5	+47
arm	10	177	70	61	+9	55	8	+47
smoke	8	248	65	96	-31	26	3	+23
wake	8	132	74	60	+14	53	30	+23
forge	7	117	54	33	+21	38	29	+9
rush	6	141	69	53	+16	43	8	+35
talk	6	79	42	73	-31	19	22	-3
call	5	168	57	32	+25	36	19	+17
say	4	216	82	86	-4	44	3	+41
enlarge	4	154	75	91	-16	26	13	+13
cry	4	119	55	57	-2	60	0	+60
import	4	100	73	92	-19	25	15	+10
explain	4	93	80	58	+22	80	23	+57
cross	3	437	66	51	+15	54	0	+54
speed	3	180	82	73	+9	20	2	+18
throw	3	165	78	30	+48	59	1	+58
arrest	3	100	88	97	-9	35	15	+20
MEAN	11	164	66	62	+4	43	12	+31

Table 1: Results for the pattern classification task

$$\text{AScore}(w,r,w') = \log \frac{\|w, r, w'\| \cdot \|\ast, \ast, \ast\|}{\|w, r, \ast\| \cdot \|\ast, \ast, w'\|} \cdot \log(\|w, r, w'\| + 1) \quad (2)$$

$$\text{Dist}(w,w') = \frac{\sum_{(tuple_i, tuple_j) \in \{tuple_w \cap tuple_{w'}\}} AS_i + AS_j - (AS_i - AS_j)^2 / 50}{\sum_{tuple_i \in \{tuple_i \cup tuple_j\}} AS_i} \quad (3)$$

$$\text{score}_{v,k} = \sum_{(w,r)} \sum_{(w',r)} \max \left(\text{Dist}_{w,w'} \cdot \frac{\sum_{(w,r,k)}}{\sum_{(w,r)}} \right) \quad (4)$$

SketchEngine. Results showed that the system beats the baseline on average and has a high precision, which makes it a potentially interesting tool for NLP applications. Various grammars or methods to generate thesaurus contexts need to be tested in order to improve the system’s recall without sacrificing precision. In the future, the system will also be analysed on a larger set of verbs.

Acknowledgements

We would like to thank Patrick Hanks, Jane Bradbury, and anonymous reviewers for their comments on an earlier draft. This work was supported by AHRC grant [DVC, AH/J005940/1, 2012-2015]. It has also been partially supported by the Ministry of Education of Czech Republic within the LINDAT-Clarin project LM2010013.

References

- Charles J. Fillmore. 1985. Frames and the semantics of understanding. *Quaderni di Semantica*, 6(2):222–254.
- Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery (The Springer International Series in Engineering and Computer Science)*. Springer.
- Patrick Hanks. 2004. Corpus pattern analysis. In *Eurolex Proceedings*, volume 1, pages 87–98.
- Patrick Hanks. 2013. *Lexical Analysis: Norms and Exploitations*. MIT Press.
- Adam Kilgarriff and Pavel Rychly. 2010. Semi-automatic dictionary drafting. In Gilles-Maurice de Schryver, editor, *Oxford Handbook of Innovation*. Menha Publichers, Kampala.
- Adam Kilgarriff, Pavel Rychly, Pavel Smrz, and David Tugwell. 2004. Itri-04-08 the sketch engine. *Information Technology*, 105:116.
- Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*. Association for Computational Linguistics.
- Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69.
- James Pustejovsky. 1995. *The Generative Lexicon*. MIT Press.
- Pavel Rychlý and Adam Kilgarriff. 2007. An efficient algorithm for building a distributional thesaurus (and other sketch engine developments). In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 41–44, Prague, Czech Republic. Association for Computational Linguistics.
- John Sinclair. 1991. *Corpus, concordance, collocation*. Oxford University Press.
- Yorick Wilks. 1975. A preferential, pattern-seeking, semantics for natural language inference. *Artif. Intell.*, 6(1):53–74.