229 Project

Reuben Cohn-Gordon

Abstract

Compositionality, that the meaning of a linguistic unit is determined by the meaning of its parts, is a core principle in the study of natural language semantics. However, many linguistic units are not compositional, such as idioms like *high time*, and multiword expressions like *hot dog*.

As such, developing methods to identify when a linguistic unit is acting compositionally, and identifying which phrases tend to be non-compositional is valuable both for computational applications and theoretical understanding.

By extending an unsupervised approach to word sense detection and induction which uses Latent Dirichlet Allocation, we develop a method to measure phrase compositionality in context and identify frequently non-compositional phrases across a corpus.

1 Introduction

Consider the sentence in (1). In order to successfully interpret the meaning of this sentence, it is necessary for a reader or hearer to know not only the meanings of the individual words it contains, but also a number of **multiword expressions**, such as *attorney general*, *look over* and *at length*.

(1) The attorney general looked over the papers at great length.

Multiword expressions are *non-compositional*, in the sense that their meanings cannot be entirely predicted from their constituent parts.

Importantly, compositionality or lack thereof is not a fixed property of a phrase. That is, some phrases, like *cold feet* or *black box* appear compositionally in some sentences and non-compositionally in others (see examples (2) and (3)). However, many phrases have a strong tendency to be either uniformly compositional or non-compositional, as in *red sofa* or *high horse* respectively.

(2) John gets cold feet when he wears these shoes.

(3) John has cold feet about the marriage.

The focus of this project is to build a system which performs the following two related tasks:

(4) Non-compositional sense detection: given a phrase in a sentence, determine whether that phrase is being used non-compositionally.

(5) Non-compositional sense induction: given a phrase in a corpus, determine how likely that phrase is to be used non-compositionally, and what it means.

Unlike typical approaches to these questions, which approach goal (4) as a supervised classification task, we propose an *unsupervised* approach, based on a probabilistic generative model. Concretely, we extend Latent Dirichlet Analysis (LDA) into a model described in section 4 in which an inference is drawn both as to the overall propensity for compositionality of the phrase (with a Beta distributed prior) and the probability of it being compositional in a given sentence.

This approach is appropriate for the task for two reasons. First, the model we propose is a natural extension of existing applications of LDA to word sense detection and induction. In this setting, the two tasks of induction and detection are handled in a unified manner by a single model. Second, Bayesian approaches such as LDA are well-suited to small data settings - humans are able to identify and learn the meanings of multiword expressions from only a few occurrences, and in corpus data, multiword expressions may occur only rarely.

2 Related Work

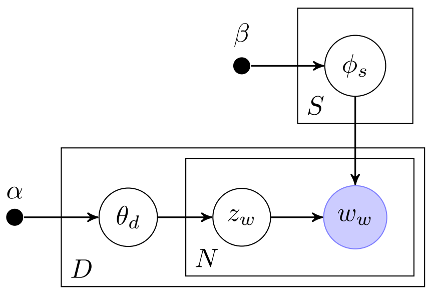
Multiword Expression Detection A common NLP task is to detect *multiword expressions*, such as *red tape* or *spill the beans*. The inability to identify multiword expressions can lead to failure in NLP tasks which require language understanding [Sag et al., 2002], and as such, their identification has been the subject of continued work in the NLP literature.

Multiword expression detection encompasses compositionality detection, since multiword expressions are non-compositional, but may include other considerations, in particular syntax. As an example, *by the by* breaks the usual syntactic rules governing the use of *by*, and is identifiable as a multiword expression for that reason. By contrast, the focus of this project is purely on the semantic phenomenon of compositionality.

Sense Induction and Detection *Word sense detection* is the task of identifying which sense of a word is active in a particular context (i.e. the surrounding text in which the word appears). Since orthographically identical strings may have more than one meaning (e.g. *bat* denotes both an animal and a man-made object), this task is an important part of NLP systems for language understanding.

Word sense detection is a classification task, requiring a set of senses to be prespecified for a given word. As such, a naturally related task is the *induction* of a set of senses. This can either be carried out using a hand-built resource, such as WordNet, or can be viewed as an unsupervised clustering problem.

3 Senses as Topics



```
1:  for i in range(K) do
2:      Choose $\phi^i_{w_1} \sim$ Dirichlet($\beta$)
3:  end for
4:  for sentence $d$ in corpus $D_{w_1}$ do
5:      Choose $\theta^d_{w_1} \sim$ Dirichlet($\alpha$)
6:      for position $j$ in $d$ do
7:          Choose a sense index $z_j \sim$ Multinomial($\theta^d_{w_1}$)
8:          Choose a word $x^j_d \sim$ Multinomial($\phi^{z_j}_{w_1}$)
9:      end for
10: end for
```

Figure 1: Plate diagram and pseudocode describing generative process of LDA.

By drawing a connection between word sense induction and detection, and topic modeling, it is possible to unite the two tasks into a single Bayesian framework. The idea is to treat word senses as topics in the sense of Latent Dirichlet Analysis Hoffman et al. [2010], and sentences containing that word as "documents". A Bayesian variant of the latter approach is implemented by Brody and Lapata [2009], in which latent Dirichlet analysis (LDA) is used to infer a set of senses for a given word from a corpus.

The generative process used here is as follows for a model with i senses: i categorical distributions over the vocabulary (each denoting a word *sense*) are drawn from a symmetric Dirichlet. For a given

```
1: Run LDA on $D_{w_1}$ and $D_{w_2}$ to obtain $\{\phi^i_{w_1}\}$ and $\{\phi^i_{w_2}\}$
2: Choose $\kappa \sim$ Beta$(\gamma, \gamma)$
3: Choose $\omega \sim$ Beta$(\delta, \delta)$
4: for i in range(K) do
5:     Choose $\phi^i_p \sim$ Dirichlet$(\beta)$
6: end for
7: for sentence $d$ in corpus $D_p$ do
8:     Choose $\lambda_d \sim$ Bernoulli$(\kappa)$
9:     if $\lambda$ then
10:        Choose $\theta^d_{w_1} \sim$ Dirichlet$(\alpha)$
11:        Choose $\theta^d_{w_2} \sim$ Dirichlet$(\alpha)$
12:        for position $j$ in $d$ do
13:            Choose $w \sim$ Bernoulli$(\omega)$
14:            Choose a sense index $z_j \sim$ Multinomial$(\theta^d_w)$
15:            Choose a word $x^j_d \sim$ Multinomial$(\phi^{z_j}_w)$
16:        end for
17:    else
18:        Choose $\theta^d_p \sim$ Dirichlet$(\alpha)$
19:        for position $j$ in $d$ do
20:            Choose a sense index $z_j \sim$ Multinomial$(\theta^d_p)$
21:            Choose a word $x^j_d \sim$ Multinomial$(\phi^{z_j}_p)$
22:        end for
23:    end if
24: end for
```

Figure 2: Pseudocode describing the generative model inherent in CLDA.

sentence (or window of text) in the training set, a categorical distribution over senses is drawn also from a symmetric Dirichlet distribution, a sense is sampled, and the value of the nth word is sampled from the sense. This constitutes a generative process for the data and allows for the inference of the various latent variables, including the categorical distributions corresponding to senses, using Monte Carlo or Variational methods.

4 Extending LDA for sense detection and induction to multiple words

In this section, we describe the model we use to perform non-compositionality detection and induction, which we term Compositional Latent Dirichlet Allocation (CLDA).

The core insight behind CLDA is that, when acting non-compositionally, a multiword expression can be treated like a single word, in terms of the generative process laid out by LDA. That is, we can detect and induce the senses of a phrase, using the same technique as for a word.

Futhermore, we include in our model a random variable determining whether a phrase is in general likely to be compositional (κ), whether it is compositional in a particular document d (λ_d), and a random variable determining how likely each of the two constituent words is to determine the makeup of the sentence if the phrase *is* compositional (ω). If it is, the generative process proceeds by treating the phrase as a single word. If it is not, the senses of the constituent words are involved.

To explain the model concretely, we first describe the generative process of the data from the latent variables. As in LDA, the model is then inverted by Bayesian inference to give posterior distributions over the latent variables. In pseudo-code, the model is as follows:

The quantities of interest that we wish to obtain are:

- κ: this represents the propensity for compositionality of the phrase
- $\{\lambda_d\}$: this represents whether the phrase is compositional when it appears in sentence d

The intuition is that if a phrase appears with words substantially different to the words which are probable under the sense distributions of either of its constituents, then it is likely to be non-compositionaly, and a separate set of senses is required to explain the data D_p.

5 Inference

While the generative model under use is fairly simple to describe and to forward sample from, performing inference is far more complex, and necessitates a special approximate inference strategy.

For standard LDA, several different inference strategies are commonly used. These include expectation propagation, variational Bayes, and Markov chain Monte Carlo (MCMC). In particular, Gibbs sampling and collapsed Gibbs sampling are standard methods used (see [Murphy, 2012] for details).

We envision that all of these techniques are applicable to CLDA. While variational methods are scalable to very large datasets, our dataset is relatively small, and we opt for MCMC, as it allows more straightforward customization of the inference algorithm.

6 Implementation

We implement the model in probabilistic programming language Gen [Cusumano-Towner and Mansinghka, 2018]. Probabilistic programming languages allow distributions to be expressed as algorithms similar to figure 2, by incorporating stochastic primitives, and provide inference strategies to be applied to these models. One motivation of this approach is the ability to rapidly prototype Bayesian models.

Gen is written in Julia, a numerical computing library well suited to writing fast inference algorithms for large datasets. It represents executions of probabilistic programs by their *trace*, i.e. a possible assignment of all the random variables involved in the model, and an associated log probability. This allows for highly customizable MCMC algorithms, where the kernel, and in particular the proposal of a new trace from an old one, can be careful controlled.

For reasons of speed, we use importance sampling. Unfortunately, this limits the quality of the model, since samples of the posterior obtains through this method are of a low quality.

As an example of the behavior of the model on toy data, suppose $D_{w_1} = [[1,1,1],[1,1]]$, $D_{w_2} = [[2,2,2],[1,1]]$ and $D_p = [[3,3,3],[3,3,3,3]]$. Note that the numbers represent the indices of words in a vocabulary. In this case, we obtain a sample of the posterior in which $\kappa = 0.106$ and $\lambda_1 = \lambda_2 = 0$. However, supposing that $D_p = [[1,1,1],[3,3,3,3]]$, we obtain $\kappa = 0.776$ and $\lambda_1 = 1 \neq \lambda_2 = 0$. This shows that the model is behaving as expected in very simple cases.

6.1 Dataset

In order to make use of CLDA, it was necessary to assemble a corpus of sentences in which various typically non-compositional phrases occur, as well as corpora of sentences containing each of the constituent words. This was achieved by constituency parsing the British National Corpus and then using tregex [Levy and Andrew, 2006], a tree based extension of regular expressions, to search for adjective noun phrases matching a number of common idioms. Before perform this tregex search, we lemmatize the leaf nodes of the trees, so that morphological variants of idioms are detectable (*hot dogs*, *spilled the beans*).

For each phrase in a set of commonly idiomatic phrases, I preprocess the associated sets of sentences containing the whole phrase, first word, and second word, by converting each sentence into a list of integer indices and removing the relevant word or phrase.

Unfortunately, we have not yet successfully implemented an MCMC inference algorithm that can handle large datasets, and as such, it was not possible to evaluate the model on its predictions, regarding goals 4 and 5.

7 Future Work

Developing a more scalable inference algorithm will be necessary in useful applications of the model to real data. In particular, the aim is to be able to validate the hypothesis that κ is a good measure of potential for non-compositionality, while λ_d is a good measure of the compositionality of a phrase in a particular sentence d. At present, attempts to implement MCMC face the problem that each update is slow, and the chain is also slow to converge.

A second direction of future work concerns the fact that LDA and CLDA assume a fixed set of senses, which contrasts with the intuition that some words and phrases should have a potentially unbounded number of senses, and that this number should be inferable from data. To address this shortcoming for word sense induction, Yao and Van Durme [2011] proposes a non-parametric extension of the LDA approach, in terms of a hierarchical Dirichlet process (HDP). In short, the Dirichlet distributions are replaced by Dirichlet processes. While samples from a size k Dirichlet distribution are distributions over sets of cardinality k for $k \in \mathcal{Z}$, samples from a Dirichlet process are distributions over $\mathcal{Z}$. In other words, a sample from the Dirichlet process is a distribution over a countably infinite set, of which some number are assigned non-zero probability.

Non-parametric extensions of LDA, in particular Hierarchical Dirichlet Processes (HDPs) provide a method to avoid prespecifying a fixed number of topics. This approach has been applied to word sense induction by Brody and Lapata [2009], and could be applied to the present task too.

8 Code

`https://github.com/reubenharry/Compositional-Latent-Dirichlet-Allocation`

References

Samuel Brody and Mirella Lapata. Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–111. Association for Computational Linguistics, 2009.

Marco Cusumano-Towner and Vikash K Mansinghka. A design proposal for gen: probabilistic programming with fast custom inference via code generation. In *Proceedings of the 2nd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pages 52–57. ACM, 2018.

Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864, 2010.

Roger Levy and Galen Andrew. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *LREC*, pages 2231–2234. Citeseer, 2006.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for nlp. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–15. Springer, 2002.

Xuchen Yao and Benjamin Van Durme. Nonparametric bayesian word sense induction. In *Proceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing*, pages 10–14. Association for Computational Linguistics, 2011.