Matching Poems in a Parallel Corpus using Concept Networks

Aubrey Barnard

Abstract

Many natural language processing approaches, and machine learning approaches in general, focus on learning "rich" hypotheses from "poor" representations. An example would be detecting humor in bag-of-words vectors. In this work, I investigate enriching the representation of text using concept networks, and apply the approach to matching poems between languages in a parallel corpus. The initial results show promise, but there is much more to investigate.

Introduction

In the quest for better natural language processing (NLP) techniques, researchers explore increasingly sophisticated computational models. Yet, the ways in which they represent text are often limited. I like to think of this situation as trying to learn "rich" hypotheses from "poor" representations. That is, typical approaches place the burden of learning and inference on the computational model rather than the text representation. These approaches seem unnatural to me when I compare them to my understanding of how the brain works, that humans learn simple concepts from rich representations. Therefore, I wished to see what would happen when trying to reverse this trend and learn a simple hypothesis from a rich representation.

The human brain encodes an incredibly rich representation of the world in its network of neurons. A network of concepts is a comprehensible and tractable abstraction of this representation. I propose representing documents in terms of concept networks as a way to enrich their representation and thereby simplify their processing.

A natural task for exploring text and concept networks is matching poems between languages in a parallel corpus. Concepts remain the same across translations, so while translation is difficult, matching concepts should be easy. Table 1 illustrates the task. This leaves the questions of obtaining a parallel corpus, creating a concept network, and translating the text into concepts.

The Parallel Corpus

I chose to work with the poems of Johann Wolfgang von Goethe (1749-1832) for several reasons. Goethe is a well-known German author whose work is old enough and popular enough to be widely available in the public domain, I know enough German to ease working with German texts, and a poem is a good unit of text to use in a matching task. A poem is often short enough to process easily while containing enough text to work with; a poem is a well-defined unit that has an obvious corresponding translation of (usually) the same length; there are many poems which makes the matching task interesting. In contrast, novels and many documents can be too long to process efficiently and are not available in sets with consistent translations. One could break them up, but this might not produce enough pieces and it might ruin the obvious correspondences between original and translation.

I was not able to find an existing parallel corpus containing Goethe's poems in German and English, so I collected my own. I obtained the original poems from a website that had a straightforward structure and appeared reasonably authentic (von Goethe 2009a). A straightforward structure was important for enabling as much automatic extraction of the poems as possible. I obtained the English translations from Project Gutenberg (von Goethe 2009b)[1] in a comprehensive and parseable format. I also obtained a third translation by using Google's language tools (Google 2009) to translate the original poems into English. This "Google-English" set enables English-English matching which provides a comparative baseline for the English-German matching.

After obtaining the source texts and encoding them as unicode, I extracted the poems common to all sources. This amounted to selecting the matching poems by hand, and having a program extract them to files. Along the way, I corrected many formatting errors and removed editorial content in the English translation. I then canonicalized the poems by expanding the most frequent contractions, converting them to lowercase, and removing all punctuation except exclamation points, question marks, and apostrophes.

The Concept Network

A concept network is an undirected graph where each node is a concept and each edge relates two concepts in some way. The idea is that the network represents information about the world through its concepts and the relationships between

[1] Alfred Edgar Bowring translated the poems in 1853 with keen attention to reproducing Goethe's poetic intent and structure. As such, they are considered the canonical translation.

Und pflanzt' es wieder	In silent corner	And planted it again
Am stillen Ort;	Soon it was set;	On quiet place;
Nun zweigt es immer	There grows it ever,	Now branches are always
Und blüht so fort.	There blooms it yet.	And so forth blossoms.
German	*English*	*Google-English*

Table 1: The last stanza of "Gefunden" ("Found") in the three languages.

them.

Ideally, language tokens and concepts belong in a single concept network. For architectural and homonymy reasons, I separated the language tokens from the concepts. I reconnected them with language-specific translation maps, which contain mappings from words in the vocabulary to the concepts they represent. In other words, I split the overall concept network idea into three pieces: the concepts, the connections of the German words to the concepts, and the connections of the English words to the concepts.

I investigated automatically creating concept networks, for example, by synthesizing word networks such as WordNet (Fellbaum 1998) and GermaNet (Hamp and Feldweg 1997). However, such networks make no provisions for translation, so while one might be able to generate concepts in each language, one would still have to unify the networks, which is equivalent to the translation problem. I concluded that automatic generation of concept networks was too large of a problem to address as part of this work. Indeed, it is a research topic unto itself as seen in (Gregorowicz and Kramer 2006). I also investigated automatically generating the translation maps using Wiktionary. While it was straightforward to download pages for words and extract translations, Wiktionary was missing translations for many of the words in my texts, and required much manual intervention. Again I concluded that automatic generation of translation maps was too large of a problem to include in this work.

In the end, it was both easier and more accurate to develop the concept network and translation maps myself. The advantage was that I could achieve consistency, focus, and tailor the structures to the task. The disadvantage was that creating these structures by hand limited their sizes, and hence the size of the vocabulary.

Approach

The first step in the approach is to represent the documents in terms of the concept network. My approach starts by creating bag-of-words vectors for the poems. The 300 most frequent words in each language comprise the vocabulary for that language. Words not in the vocabulary are counted in an "unknown" bucket at the end of the vector.

Then my approach converts the bag-of-words vectors into activations of the concepts in the network. This transforms the representation of the poems from words to concepts, hopefully enriching the information contained in the representation. An activation of the network is a vector of real values, where each value represents how much a particular concept was activated by the text. (Values for the language tokens are not included.) To activate the network, my approach propagates a word count to its neighboring concepts, and then to that concept's neighbors, and so on recursively. At each propagation, the value is divided equally among the neighbors, and only that fraction is propagated to that neighbor. Each concept node accumulates the values propagated to it. Propagation loops are allowed, which seems natural because similar concepts reinforce each other. For my experiments, I limited the propagation to five levels deep due to the exponentially increasing number of nodes in the propagation tree. My approach repeats the propagation process for each word count in the bag-of-words vector including the "unknown" bucket. The result is an activation vector, the accumulated values for each concept node.

The next step is matching poems. Each English poem matches the nearest German poem. My approach uses a simple ("poor") hypothesis and matches the poems with the minimum distance between activation vectors. My approach uses the L2 (Euclidean) and L1 norms as distances, although any distance metric could be used.

The main difference between my concept network approach and the one used for the comparative baseline is that the bag-of-words vectors are not converted into concept activation vectors. That is, each English bag-of-words matches the nearest Google-English bag-of-words. There is a single vocabulary, the 300 most frequent words in the combined English and Google-English poems plus "unknown." There are also some different distances. One is cosine similarity formulated as a distance,

$$d_{cosine}(\mathbf{x}_1, \mathbf{x}_2) = 1 - \frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{\|\mathbf{x}_1\|\|\mathbf{x}_2\|},$$

and another is Hamming distance,

$$d_{Hamming}(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=1}^{V} |\text{sign}(x_{1,i}) - \text{sign}(x_{2,i})|.$$

Evaluation

Table 2 contains the matching accuracies. The accuracy is how many English poems correctly match to their German or Google-English counterparts. The concept network approach performs respectably, but the Hamming and L1 norm distances surprisingly perform the best despite the many flaws in the Google-English poems.

There are many ways to improve the performance of the concept network approach. I think improving the vocabulary would result in the largest gains. With only 300 words, the vocabularies are weak. In particular, they lack nouns which are important for matching. Removing stop words and stemming would help the vocabulary by focusing it. Automating

Poem Set	*Method*	*Accuracy*	
Google-En	Cosine	53/127	41.7%
Google-En	Euclidean	58/127	45.7%
German	Concept-Euclidean	71/127	55.9%
German	Concept-L1Norm	74/127	58.2%
Google-En	Hamming	91/127	71.7%
Google-En	L1Norm	101/127	79.5%

Table 2: Matching accuracy by poem set and method.

the generation of the concept network and translation maps would help the vocabulary by expanding it. Additionally, I could experiment with how to compute and compare activation vectors. For example, a time-based approach seems promising because it would incorporate the progression of concepts in the text.

Conclusion

It appears that a simple, distance-based approach works well for matching poems between languages in a parallel corpus, and that concept networks have the potential to enhance the matching process with enriched information. Certainly their potential extends beyond poem matching to real-world applications such as information retrieval and text synthesis. However, achieving real-world performance with concept networks remains an open investigation.

References

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Google. 2009. Google language tools.

Gregorowicz, A., and Kramer, M. A. 2006. Mining a large-scale term-concept network from Wikipedia. Technical report, Mitre.

Hamp, B., and Feldweg, H. 1997. Germanet – a lexical-semantic net for german. In *In Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, 9–15.

von Goethe, J. W. 2009a. Johann Wolfgang von Goethe: Gedichte. `http://www.wissen-im-netz.info/literatur/goethe/gedichte/index.htm`. Published by Jürgen Kühnle.

von Goethe, J. W. 2009b. The poems of Goethe. `http://www.gutenberg.org/etext/1287`. Translated by Edgar Alred Bowring. Published by Project Gutenberg.