Text Classification Using Graph-Encoded Linguistic Elements

Kevin R. Gee and Diane J. Cook
Department of Computer Science and Engineering
The University of Texas at Arlington
Arlington, TX 76019
{geek,cook}@cse.uta.edu

Abstract

Inspired by the goal to more accurately classify text, we describe an effort to map tokens and their characteristic linguistic elements into a graph and use that expressive representation to classify text phrases. We outperform the bag-of-words approach by exploiting word order and the semantic and syntactic characteristics within the phases. In this study, we map tagged corpora into a placeholder graph structure and classify the phrases within, using the cross-dimensional linguistic characteristics of each token. Finally, we present heuristics for use in applying this method to other corpora.

Introduction

In the field of natural language processing, there is a need to analyze to analyze a language data that surpasses the traditional bag-of-words approach, so named for the practice of treating each word token as a distinct entity whose value depends solely on its presence in the selection, with no consideration given to any other inherent characteristic or attribute. The obvious preference is to analyze these inherent attributes of each word token, such as the part of speech, its semantic relationship to other words in the sentence, or its role in projecting meaning.

For instance, the bag-of-words approach would evaluate the token "say" the same whether it be used as a verb to utter or pronounce ("say it out loud"), a verb to state an opinion ("I say we leave now!"), a verb to recite ("say grace"), a noun indicating an opportunity to speak ("have one's say"), a discourse maker ("say, that's a nice car"), or even used in a proper noun or title ("Say Say Say"). While this approach works for many tasks based on a simple token search, it is inadequate for tasks requiring the use of linguistic meaning.

Even when linguistic information is included, it is a fairly routine task to identify commonly repeating patterns that use a single semantic dimension. For example, a corpus might have a relatively high number of sentences that all begin with this sequential part-of-speech string that does not appear in other corpus types:

Adverb → Comma → Personal Pronoun → Verb

What is more difficult, and interesting, is the identification of patterns that cross dimensions. These dimensions might include the specific term itself ("bank"), the semantic tense of the term (bank vault, river bank, or an aviation term), the word class ("bank" as an institutional building, like a church), the part-of-speech or its general type (plural personal pronoun versus any general pronoun) or discourse markers.

While natural language grammars allow for some variability in syntactic ordering, some linguistic tasks, like stylometry, depend on the order in which specific tokens or general classes of tokens are used.

One common approach to mining information from components with multiple dimensions is to generate a set of all tuples and mine for common patterns using an a priori approach (Agrawal and Srikant 1994, WordSmith 1996). In a linguistic approach where words have multiple dimensions, the sheer number of permutations makes this approach undesirable.

This paper outlines a method by which sequences of words can be analyzed for common structural similarities and classified based on those similarities, instead of relying solely on a bag-of-words approach. Quantitative results are presented both on artificially-generated and naturally-occurring language data. The research shows that graphs can correctly be discovered that include multiple semantic dimensions (parts of speech, word types, specific tokens), and several sample graph structures are presented.

Graph Processing

The core algorithms to discover and classify the frequently-occurring patterns in the language data are supplied by the SUBDUE system (SUBDUE 2004), which can detect repetitive patterns, or substructures, within graphs. We define a graph as a set of labeled vertices and labeled edges between those vertices, where the labels need not be distinct. A substructure of the graph is a connected sub-graph.

Note that this paper's focus is not to validate SUBDUE or its approach. Enough studies have been performed on disparate datasets to demonstrate its value (Cook and Holder 2000, Coble and Cook 2003), including studies

involving text processing (Aery and Chakravarthy 2005). For this particular research, SUBDUE and its algorithms are used as the graph-processing backend; however, any graph-based pattern detection algorithm would perform equally well.

SUBDUE maintains an ordered set of discovered substructures called the parent list; initially, this list simply holds a single-vertex substructure for each distinct vertex label. These simple substructures are removed from the list; their extensions in the main graph are produced by adding either a new vertex and the corresponding edge or just an edge, and then inserted into the parent list. As new substructures are generated, a second list maintains the best substructures discovered. After completing this process, the substructure with the top utility is reported. The graph may then be compressed by replacing each instance of the top substructure with a new, unique, vertex to represent that larger substructure.

Two common graph evaluation techniques are Minimum Description Length (MDL) and Set Cover (Cook and Holder 2000). The MDL heuristic measures the minimum description length (or simply "description length"), which is the lowest number of bits needed to encode a piece of data; SUBDUE approximates this value for any given graph. Using this heuristic, the best substructure is that which minimizes the following value:

$$\text{value}(S,G) = \text{DL}(G)/(\text{DL}(S) + \text{DL}(G \mid S))$$

where G is the entire graph, S is a particular substructure, and DL is the description length in bits, and (G|S) is G compressed with S.

When evaluating graphs using a set cover function, the value of a substructure S is computed as the number of positive examples containing S plus the number of negative examples not containing S, this quantity divided by the total number of examples (SUBDUE 2004). The resulting compression removes all positive examples in the graph containing S.

Here is a simple example of how SUBDUE works, given the graph shown in Figure 1.

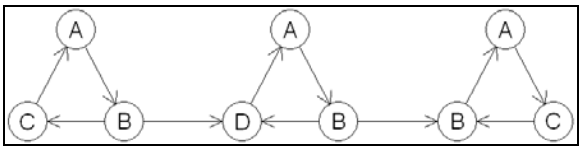
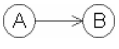
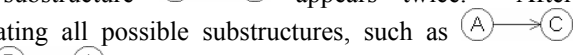
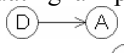


Figure 1. Example graph.

The substructure  appears twice. After evaluating all possible substructures, such as  and  for example, SUBDUE will rank this substructure () the best. Before the next iteration, SUBDUE will compress the graph by replacing all instances of this substructure with a new vertex. Like other graph-based algorithms, SUBDUE time complexity is exponential in the worst case, but can be reduced to polynomial time in practice (SUBDUE 2004).

As noted previously, this work does not intend to re-validate SUBDUE or the concept of processing text via graphs. The approach to detecting linguistic patterns should work with any graph processing algorithm.

Experiments

We encode text phrases as graphs, encapsulating different combinations of linguistic dimensions such as word order, part-of-speech syntax, and semantic elements, with the aim to more accurately classify the phrases.

We demonstrate these results using two artificial corpora, Word Order and Garden Path, and two natural corpora, ACES and Physics. The artificial corpora were selected for their ability to demonstrate how the approach is working for specific cases. A basic placeholder graph structure is used, and five-fold validation testing is used to test each corpus.

Graph Structure

This approach models language using a "placeholder" graph, where a vertex with a generic label is used to represent each word, with edges leading out from the placeholder to vertices representing the dimensions (each placeholder has the same label). Figure 2 is an example of two words, t1 and t3, encoded with the following dimensions:

- Tokens t1 and t3 (the words themselves)
- The respective parts of speech t2 and t4
- Their status as either a content or function word

Each dimension appears as a vertex with an edge from the placeholder representing the word (the "Plch" vertex at the top of the diagram) and with another vertex to the placeholder for the succeeding token (the "Plch" vertex in the middle and at the bottom of the figure). Other desired characteristics, such as a proper noun or a particular word class would also be represented as vertices, with edges linking the vertex for the dimension to the preceding and succeeding placeholders.

One crucial feature of the graph is the edge that exists between each placeholder (see the left-most edge in the

diagram). With the edge in place, a sequence of words X, Y, and Z can reveal patterns involving only dimensions of X and Z, without any regard for any dimension of Y. Without the edge, characteristics of Y must be taken into account, an unnecessary requirement.

The edge from each dimension to the succeeding placeholder is necessary as well, despite effectively doubling the number of edges need for the graph (resulting in a four-fold increase in processing time in some tests). Without these additional edges, it is very difficult, after vertices have been compressed, to determine how vertices compressed in one iteration link back to other vertices compressed in previous iterations. This is compounded in that the edge labels contain essential information to analyzing the final data. Note that the problem of losing information through iterative compression is not eliminated with these extra edges to the next placeholder vertex, but it is greatly reduced in practice.

Succeeding words are chained to the graph by linking vertices representing their dimensions to the last placeholder, and by adding a final placeholder. Any study that intends to include possible patterns that cross sentence or paragraph boundaries should include sufficient end-of-sentence markers in the chain; punctuation with appropriate tags usually suffices. If inter-sentence patterns are not relevant, then it was observed during testing that lightly better runtime exists if each distinct semantic unit (sentence, phrase, etc.) was broken up into separate chains in the same graph.

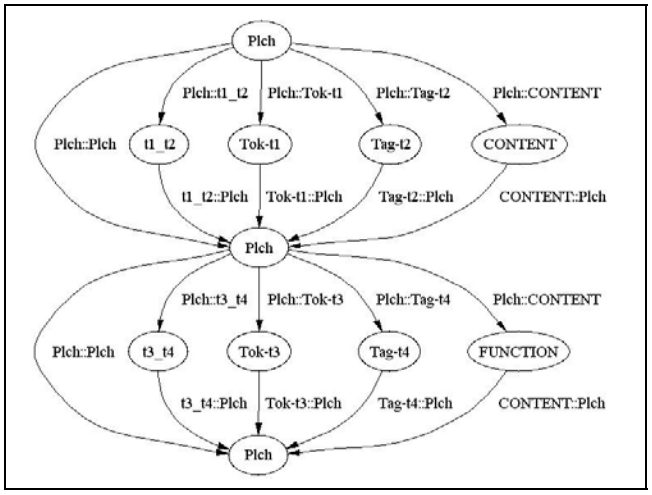


Figure 2. Graph structure of content word *t1* and function word *t3*, with tags *t2* and *t4*, respectively, their placeholders, and the next placeholder.

Dimensions Used

These linguistic dimensions for each word were used in graph generation:

- The token (e.g., "Models").
- The tag ("NN1" is a singular noun, for instance).
- The actual tagged token itself ("Models_NN2").
- Whether or not the token is a content word, a function word (Au-Yeung and Howell 1999), or punctuation.
- Whether or not the word is a hyphenate ("Event-Driven"), a technical, scientific, or invented word ("eMailSift"), a proper noun or a number.
- A generic tag, such as "Noun", since most tagsets contain very granular tags that are roughly equivalent (for example, NN0, NN1, NN2, and UNC are valid tags for nouns in the C5 tagset (Leech 1997)).

Numbers were considered to be content words. While pronouns are commonly classified as function words (Au-Yeung and Howell 1997), they are considered content words for this study, since normally it would not be expected to find pronouns in a presentation title. To distinguish between tag classes and word classes, the latter used capital letters and the former enclosed the label between parentheses. Also, the token "not" (or "n't") was classified as a function word for the purposes of this study.

It should be noted that *any* semantic or syntactic element could be graphed and used for classification, including discourse markers, noun phrases, coreference relationships, etc. Which linguistic dimensions are graphed is a feature selection issue and can be altered when needed.

Classification Testing

Classification testing was performed on all corpora using four different types of structures, using the set cover graph evaluation method:

- A bag-of-words approach (in the results as "BOW")
- A directed graph reflecting the word order of the tokens in each sample ("Ordered")
- A directed graph encapsulating the tag and the word class structures, without using the terms themselves ("Structure")
- A directed graph with the tokens, their word order the tokens, and their syntactic/semantic structure ("O+S")

Following our methodology above, all reported numbers represent averages across five-fold validated runs.

A separate graph was created for each individual subclass, where members of that class were marked as positive

examples and all members of all other classes from that corpus were marked as negative examples.

Corpora

We demonstrate these results using two artificial corpora, Word Order and Garden Path, and two natural corpora, ACES and Physics. The artificial corpora were selected for their ability to demonstrate how the approach is working for specific cases.

Word Order. This artificial corpus was constructed primarily to validate the accuracy of the approach, by gauging the effects on a corpus where the tokens in each class are identical, both in frequency and distribution. It consists of two classes, the first containing a series of sentences generated from all possible permutations of these elements:

(((a|the) (boy|girl|child)) | (he|she|they))
(bought|purchased|acquired|sold)
(toys|candy|food|soda)
(yesterday|today)

The phrases included samples like these:

- a boy bought candy today
- the girl purchased food yesterday
- she acquired toys today

The second class consisted of the same sentences but reordered into non-canonical word order, using permutations of this order:

(yesterday|today)
(((a|the) (boy|girl|child)) | (he|she|they))
(toys|candy|food|soda)
(bought|purchased|acquired|sold)

The phrases included samples like these:

- today a boy candy bought
- yesterday the girl food purchased
- today she toys acquired

A bag-of-words approach would find that the tokens in each class are completely identical, and any classification for either class would thus accept all examples from both classes, yielding an error rate of 50% (since the number of true positives would equal the number of false positives). Encoding word order into the graph causes the examples to be classified correctly, as is shown in the next section.

Garden Path. This artificial corpus was constructed using so-called "garden path" parsing problems, where a phrase consists of polysemous words, yielding two very different meanings depending on how the tokens are tagged when parsed (Dougherty et al 2001). The different meanings for each phrases were split into different classes. Examples of these phrases are:

- fruit flies like an *orange* (*flies* or *like* can be the verb)
- the historian knew some *rebel* (*rebel* is a noun or verb)
- the FBI discovered some *record* (*record* is a noun or verb)

A bag-of-words approach would yield an error rate of 50%, for the same reason as the Word Order corpus. Additionally, a directed graph encoding just the tokens in their occurring word order would also yield an error rate of 50% (with the number of true positives equaling the number of false positives), since the word order is identical in each class. Encoding the semantic structure into the graph causes the examples to be classified correctly, as is shown in the next section.

ACES. This corpus is taken from the set of presentation titles from the ACES 2004 conference (ACES 2004) and has nine classifications corresponding to titles originating from different departments. Each classification has its own set of unique terms and the samples within have their own particular syntactic style. The average length of each sample was about 13 tokens.

This corpus will illustrate the effects in a multiclass environment where the difference in the amount of positive and negative examples for each class is pronounced.

Physics. This corpus was constructed from a set of Physics paper titles taken from LANL (also used in Zelikovitz 2004). This corpus is of interest because the topicality of the different classes is very similar in nature, making the bag-of-words approach very difficult. The average length of each sample was about 8 tokens.

Results

Table 1 contains the error rates from applying each approach against each class in each corpus. $ACES^E$ is a subset of the ACES corpus that benefited from the approach. The best test results for each corpus are in **bold**.

corpus	BOW	Ordered	Structure	O+S
W.O.	0.500	**0.000**	**0.000**	**0.000**
Garden	0.500	0.500	**0.000**	**0.000**
ACES	**0.125**	0.134	0.169	0.128
$ACES^E$	0.099	0.088	0.125	**0.078**
Physics	0.500	0.448	**0.218**	**0.218**

Table 1. Error rates for each method, for each corpus.

One immediate observation springs out. For the full ACES corpus, as a general rule, the bag-of-words approach outperforms the other three. This is primarily due to two factors: The lexica for some classes were very distinct and as such could be classified solely by a BOW approach, and with nine classes of roughly equal size, the number of negative examples outpaces the positive examples, making it harder for structural tests to have an impact. These phenomena, plus results for each class, are discussed in the following sections.

Class Error Rates Per Corpus

Word Order. As mentioned previously, implementing a directed graph to reflect the order in which the words appear in context, classifies correctly a corpus consisting of classes in which the terms are identically distributed but in a different order ("Ordered", in Table 2). Since all terms are equally distributed, using a directed graph representing the syntactic structure (which would also be different in each class) also achieve perfect rates.

One pattern discovered in the course of classification was that all instances of the cwo class begin with an article (*the* or *a*), which under the Set Cover algorithm is sufficient to classify all positive and negative samples in this case.

Class	BOW	Ordered	Structure	O+S
cwo	0.500	**0.000**	**0.000**	**0.000**
ncwo	0.500	**0.000**	**0.000**	**0.000**

Table 2. Error rates for each class in Word Order.

Garden Path. The "Ordered" graph performed no better than a bag-of-words approach, since the terms are identical in each class and appear in the same order. However, introducing syntactic tags in "Structure" and in the combined "O+S" graphs correctly classified every sample. The set cover patterns discovered consisted precisely of the tag differences between each class.

Class	BOW	Ordered	Structure	O+S
v1	0.500	0.500	**0.000**	**0.000**
v2	0.500	0.500	**0.000**	**0.000**

Table 3. Error rates for each class in Garden Path.

ACES. As discussed, with four of the nine classes in ACES did our graphed approach outperform a bag-of-words approach, and with three other cases, performed equally well or within a tenth of a percent (Table 4).
In almost every case, "Structure" did not fare as well as "Ordered"; this is to be expected since the "vocabulary" of the "Structured" graphs (consisting of word classes, tags, and tag classes) is not as diverse from class to class as the "Ordered" graphs, which actually use the lexicon of the class. The exception is the *nurs* class, from which a highly detailed graph was extracted with very diverse semantic elements linked together.

Class	BOW	Ordered	Structure	O+S
biome	**0.076**	**0.076**	0.116	**0.076**
chem	0.112	**0.107**	0.191	**0.107**
cse	**0.204**	0.226	0.239	0.214
ee	**0.109**	**0.109**	0.209	**0.109**
kine	0.087	**0.037**	**0.037**	**0.037**
ling	0.102	**0.096**	0.180	**0.096**
nurs	0.087	0.107	**0.059**	**0.059**
phys	**0.081**	0.106	0.086	0.106
psych	**0.141**	0.184	0.230	0.184

Table 4. Error rates for each class in ACES.

Subsequent analysis of each class yielded these common characteristics:

- Our method worked better in classes with a higher frequency of function words and punctuation (and consequently, a lower frequency of content words) than the other classes.
- Our method worked better in classes with a higher frequency of specific word types (like hyphenated adjectives) than other classes.
- The bag-of-words method worked better in classes with a higher concentration of unique content words, particularly technical terms.

Class	BOW	Ordered	Structure	O+S
chem	0.112	**0.107**	0.191	**0.107**
kine	0.087	**0.037**	**0.037**	**0.037**
ling	0.102	**0.096**	0.180	**0.096**
nurs	0.087	0.107	**0.059**	**0.059**

Table 5. Error rates for the classes in ACES^{E}.

Physics. The "Structure" graph, either by itself or combined with "Ordered", produced significantly superior results (Table 6).

Class	BOW	Ordered	Structure	O+S
aph	0.505	0.445	**0.282**	**0.282**
cmd	0.495	0.450	**0.153**	**0.153**

Table 6. Error rates for each class in Physics.

Some observations:

- In the aph class, 26% of the tokens in the positive examples were unique to that class; in the cmt class, about 21% were unique to cmt.
- There was a relatively high degree of function words given the short textual nature of the samples (27% for aph and 25.5% for cmt).

Additional Analysis

If the lexical choice is highly specialized from classification to classification, then a simple bag-of-words approach is very accurate and providing a graph with ordered tokens or semantic structure offers no or little improvement. As an example, see the classes from ACES that did not benefit from the semantic structure.

If the lexical choice is identical from classification to classification, but the structure is drastically different from classification to classification (as seen in Word Order or Garden Path), then the incorporation of word order and semantic structure is a great benefit.

Best results from our semantic additions to the graph structure are obtained when:

- Lexical diversity between classes is relatively low (see Physics)
- The terms used appear in small clusters of terms that appear in other classifications (but do not comprise an entire example)
- When there are similar semantic structures appearing in separate classifications, yet use different terms within the structure (e.g., words from a common hierarchical class, or plural forms of the same token)

Conclusions

This paper outlines a method by which we mapped short text samples into a graph structured that encoded their work order as well as other syntactic and semantic characteristics crossing multiple dimensions. We were then able to classify examples from several corpora, outperforming the common bag-of-words approach.

Analysis shows that this method is conducive to corpora with a varied distribution of tokens and complex syntactic structures; it is not suited for corpora with highly specialized lexicons.

Further research will add other features into the graph structure to solve specific linguistic problems, as well as determine which features provide optimal results to help reduce the scope of the graph.

References

1. ACES-2004. Annual Celebration of Excellence by Students in Graduate Research and Undergraduate Research and Creative Activity. The University of Texas at Arlington, Arlington, TX. 2 April 2004.
2. Agrawal, R., Srikant, R. Fast Algorithms for Mining Association Rules in Large Databases. VLDB 1994: pp. 487-499.
3. Arey, M. and Chakravarthy, S. InfoSift: Adapting Graph Mining Techniques for Text Classification. Proceedings of the Eighteenth International FLAIRS Conference, 2005.
4. Au-Yeung J., Howell, P. Introduction to Content/Function Words. University College London 1999, 2000. http://www.speech.psychol.ucl.ac.uk/training2/intro.html.
5. Cook, D.J. and Holder, L.B. Graph-Based Data Mining. IEEE Intelligent Systems, 15(2), 2000, pp. 32-41.
6. Dougherty, R., Ferrari-Bridge, F., Dyer, L. INTEX Solves Pronunciation and Intonation Problems in Text to Speech Reading Machines. Joint International Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing, 2001. http://www.nyu.edu/its/humanities/ach_allc2001/papers/dougherty/.
7. Leech, G. A Brief Users' Guide to the Grammatical Tagging of the British National Corpus, 1997. http://www.natcorp.ox.ac.uk/what/gramtag.html.
8. Noble, C.C. and Cook, D.J., Graph-Based Anomaly Detection, Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003.
9. "say". http://dictionary.reference.com/search?q=say .
10. SUBDUE: Knowledge Discovery in Structural Databases. 2004. http://ailab.uta.edu/SUBDUE.
11. WordSmith Tools. Oxford University Press, 1996. http://www.lexically.net/wordsmith/index.html.
12. Zelikovitz, S. Transductive LSI for Short Text Classification Problems, Proceedings of the Seventeenth International FLAIRS Conference, 2004.