

Correction to Results in Arora et al. (2010)

A problem was discovered in the frequent subgraph mining package (gSpan). It gives inaccurate results for directed graphs. We recreated our annotation graphs as undirected graphs and reran the experiments. This document is a revised write-up for the experiment and results section of the paper.

1 Data and Experimental Setup

Data: For our experiments, we used one of the movie review datasets provided by Pang and Lee. (2005)¹. This dataset consists of 10662 snippets/sentences from the Rotten Tomatoes website², with an equal number of positive and negative sentences (5331 each). This dataset is different from the more common movie review dataset from Pang and Lee. (2005) which consists of full reviews. The snippets dataset was created and used by Pang and Lee. (2005) to train a classifier for identifying positive sentences in a full length review. The reason for not using the full movie review dataset and instead using the movie snippets data was to simplify the annotation graph representation for our first experiment, as snippets are much shorter (about 1 sentence) than the movie review documents (about 10-15 sentences). We use the first 8000 (4000 positive, 4000 negative) sentences as training data and evaluate on remaining 2662 (1331 positive, 1331 negative) sentences. We added part of speech and dependency triple annotations to this data using the Stanford parser (Klein and Manning, 2003).

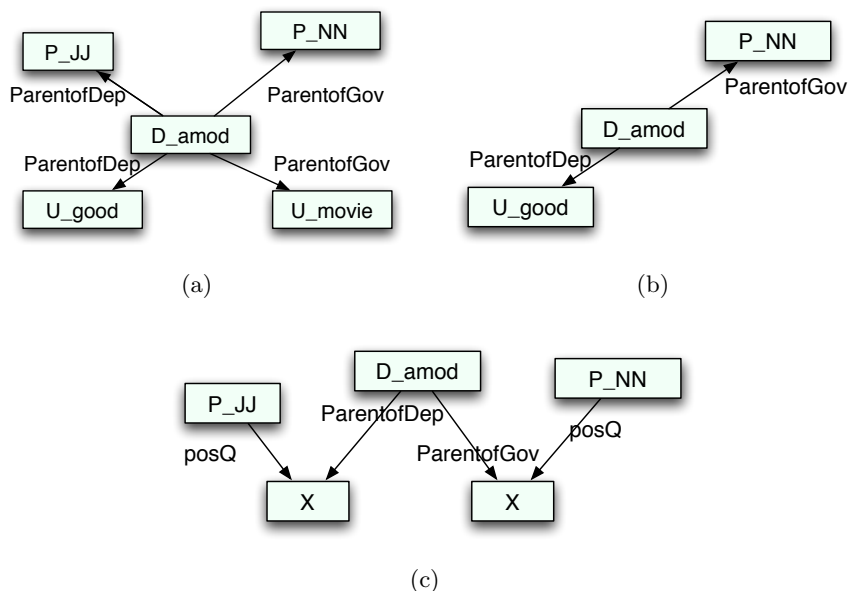


Figure 1: Annotation graph and a feature subgraph for dependency triple annotation “amod_good_camera”. (c) shows an alternative representation with wild cards

Annotation Graph: For the annotation graph representation, we used *Unigrams (U)*, *Part*

¹<http://www.cs.cornell.edu/people/pabo/movie-review-data/rt-polaritydata.tar.gz>

²<http://www.rottentomatoes.com/>

of Speech (P) and Dependency Relation Type (D) as labels for the nodes, and *ParentOfGov* and *ParentOfDep* as labels for the edges. Following the work in Matsumoto et al. (2005) we filter the unigrams that occur only once in the training data. We also use stop word lists for unigrams, part of speech and dependency relations. For unigrams, we used the stop word list from Manning et al. (2008) (with one modification: removed ‘will’, added ‘this’). For POS, we used the stop word list from Matsumoto et al. (2005) (with one modification: removed IN). For dependency relations, we used a small stop list consisting of the following relations: {det, predet, preconj, prt, aux, auxpas, cc, punct, complm, mark, rel, ref, expl}.

For a dependency triple such as “amod_good_movie”, five nodes are added to the annotation graph as shown in Figure 1a. *ParentOfGov* and *ParentOfDep* edges are added from the dependency relation node D_amod to the unigram nodes U_good and U_movie . These edges are also added for the part of speech nodes that correspond to the two unigrams in the dependency relation, as shown in Figure 1a. This allows the algorithm to find general patterns, based on a dependency relation between two part of speech nodes, two unigram nodes or a combination of the two. For example, a subgraph in Figure 1b captures a general pattern where *good* modifies a noun. This feature exists in “amod_good_movie”, “amod_good_camera” and other similar dependency triples. This feature is similar to the dependency back-off features proposed in Joshi and Rosé (2009).

Classifier: For our experiments we use Support Vector Machines (SVM) with a linear kernel. We use the SVM-light³ implementation of SVM with default settings.

Parameters: The *gSpan* algorithm requires setting the minimum support threshold (*minsup*) for the subgraph patterns to extract. Support for a subgraph is the number of graphs in the dataset that contain the subgraph. We experimented with several values for minimum support and *minsup* = 3 gave us the best performance. We report results for *minsup* = 3 and *minsup* = 5.

Feature Configurations: The dataset we used for our experiments was created and used by Pang and Lee. (2005) to train a sentence level classifier for identifying positive sentences in review documents. They did not report any performance results for the sentence classifier. To the best of our knowledge, there is no other supervised machine learning result published on this dataset. We compare the following feature configurations in our experiments:

1. *Unigram-only:* In sentiment analysis, unigram-only features have been a strong baseline (Pang et al., 2002; Pang and Lee, 2004). We only use unigrams that occur in at least two sentences of the training data same as Matsumoto et al. (2005). We also filter out stop words using a small stop word list⁴.

Following configurations combine unigrams and subgraph features and use a particular feature selection approach:

2. χ^2 *feature selection:* For our training data, after filtering infrequent unigrams and stop words, we get 8424 features. Adding subgraph features increases the total number of features to 44,161, a factor of 5 increase in size. Feature selection can be used to reduce

³<http://svmlight.joachims.org/>

⁴<http://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html> (with one modification: removed ‘will’, added ‘this’)

this size by selecting the most discriminative features. χ^2 feature selection (Manning et al., 2008) is commonly used in the literature. We compare two methods of feature selection with χ^2 , one which rejects features if their χ^2 score is not significant at the 0.05 level, and one that reduces the number of features to match the size of our feature space with GP.

3. *Feature Subsumption (FS)*: Following the idea in Riloff et al. (2006), a complex feature C is discarded if $IG(S) \geq IG(C) - \delta$, where IG is Information Gain and S is a simple feature that *representationally subsumes* C , i.e. the text spans that match S are a superset of the text spans that match C . In our work, complex features are subgraph features and simple features are unigram features contained in them. For example, $(D_amod)_Edge_ParentOfDep_ (U_bad)$ is a complex feature for which U_bad is a simple feature. We tried same values for $\delta \in \{0.002, 0.001, 0.0005\}$, as suggested in Riloff et al. (2006). Since all values gave us same number of features, we only report a single result for feature subsumption.
4. *Correlation (Corr) based feature selection*: As mentioned earlier, some of the subgraph features are highly correlated with unigram features and do not provide new knowledge. A correlation based filter for subgraph features can be used to discard a complex feature C if its absolute correlation with its simpler feature (unigram feature) is more than a certain threshold. We use the same threshold as used in the GP criterion, but as a hard filter instead of a penalty.
5. *Genetic Programming (GP) based feature combinations*: We extend the genetic programming approach in Mayfield and Rosé (2010) to our subgraph features. For our functions we use the same boolean statements, AND and XOR, while our terminals are selected randomly from the set of all unigrams and our newly extracted subgraph features. For feature construction, we divide our training data in half, and train our GP features on one half of this data. This is to avoid overfitting the final SVM model to the GP features. In a single GP run, we produce one feature to match each class value. For our sentiment classification task, a feature is evolved to be predictive of the positive instances, and another feature is evolved to be predictive of the negative documents. We repeat this procedure a total of 15 times (using different seeds for random selection of features), producing a total of 30 new features to be added to the feature space. We use the same fitness function as Mayfield and Rosé (2010):

$$\text{Fitness} = F_{\beta} + PP + CC \tag{1}$$

where F is the fitness function defined based on precision and recall of the feature for the positive and negative examples. PP is the parsimony pressure penalty to avoid overly large and complex trees. CC is a correlation penalty which penalizes correlation between the feature being constructed and the subgraphs and unigrams.

The tree in Figure 2 is a simplified example of our evolved features. It combines three features, a unigram feature ‘too’ (centre node) and two subgraph features: 1) the subgraph in the leftmost node occurs in collocations containing “more than” (e.g., “nothing

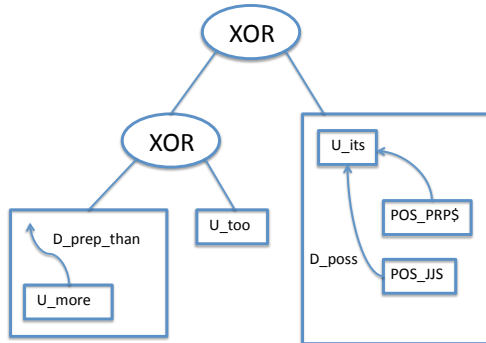


Figure 2: A tree constructed using subgraph features and GP (Simplified for illustrative purposes)

more than” or “*little more than*”), 2) the subgraph in the rightmost node occurs in negative phrases such as “*opportunism at its most glaring*” (JJS is a superlative adjective and PRP\$ is a possessive pronoun). A single feature combining these weak indicators can be more predictive than any part alone.

For genetic programming we used the ECJ toolkit⁵. We used the same parameter settings as described in Mayfield and Rosé (2010), which were tuned on a different dataset⁶ than one used in this work, but it is from the same movie review domain. We also consider one alteration to their setting. Mayfield and Rosé (2010) set the β ratio for F -measure to $\frac{1}{6}$ which gives 6 times more weight to precision. Since we are introducing a large number of features to our feature space through subgraphs and we only create 30 new GP feature combinations from all the subgraph and unigram features, we believe that a stricter constraint must be placed to find the most precise subset of feature combinations. We tried a few other values of the β parameter and $\frac{1}{15}$ gave us the best performance.

2 Results and Discussion

In Table 1 we present our results for using frequent subgraph features for sentiment classification. In sentiment classification, bag-of-words has been a very strong baseline (Pang et al., 2002; Pang and Lee, 2004) and linguistic features have not shown much substantial benefit over the unigrams. Note that the goal of this research is not to show that the subgraph features outperform unigrams on a given sentiment classification task. But rather the goal is to demonstrate that linguistic patterns relevant for a classification task that were undiscovered earlier or were hand crafted can now be automatically discovered as subgraphs from the annotation graphs without additional human effort.

One reason we believe for linguistic features not giving much benefit over unigrams in sentiment domain is sparsity. Movie reviews are written by different users who may have different writing styles. Also, movie reviews are very informal, reviewers use incomplete sentences, sarcastic language, etc. Table 2 presents some of the hard examples for an algorithm to capture. For a pattern to be discovered by a machine learning algorithm, it must be repeated in the data at least a few times and be associated with one class more than the

⁵<http://cs.gmu.edu/~eclab/projects/ecj/>

⁶Full movie review data by Pang et al. (2002)

SNo.	Settings	minSup	#Features	Acc.	Δ
1	Uni	-	8424	75.66	-
2	Uni + Sub	5	575,200	66.08	-9.58
3	Uni + Sub	3	7,019,792	64.58	-11.08
4	Uni + Sub, χ^2 sig.	5	172,338	68.82	-6.84
5	Uni + Sub, χ^2 sig.	3	2,828,488	68.52	-7.14
6	Uni + Sub, χ^2 size	5	8454	68.44	-7.22
7	Uni + Sub, (FS)	5	25,895	71.04	-4.62
8	Uni + Sub, (Corr)	5	482,073	64.65	-11.01
9	Uni + GP (U) ‡	-	8454	75.96	0.36
10	Uni + GP (U) †	-	8454	76.11	0.45
11	Uni + GP (U+S) ‡	5	8454	76	0.34
12	Uni + GP (U+S) †	5	8454	76.6	0.94
13	Uni + GP (U+S) ‡	3	8454	76.07	0.41
14	Uni + GP (U+S) †	3	8454	76.82	1.16

Table 1: Experimental results for feature spaces with unigrams, with and without subgraph features. Feature selection with 1) fixed significance level (χ^2 sig.), 2) fixed feature space size (χ^2 size), 3) Feature Subsumption (FS) and 4) Correlation based feature filtering (Corr)). GP features for unigrams only $\{GP(U)\}$, or both unigrams and subgraph features $\{GP(U+S)\}$. Both the settings from Mayfield and Rosé (2010) (‡) and more weight on precision ($\beta = \frac{1}{15}$) (†) are reported. *#Features* is the number of features in the training data. *minSup* is the minimum support threshold for frequent subgraph mining. *Acc* is the accuracy and Δ is the difference from unigram only baseline. Best performing feature configuration is highlighted in bold which is marginally significantly better than unigram based on 90% confidence interval calculated using one-way anova.

Movie Review Snippet	Class (POS/NEG)
the importance of being earnest , so thick with wit it plays like a reading from bartlett’s familiar quotations	POS
the events of the film are just so weird that I honestly never knew what the hell was coming next	POS
in the poor remake of such a well loved classic, parker exposes the limitations of his skill and the basic flaws in his vision	NEG
initial strangeness inexorably gives way to rote sentimentality and mystical tenderness becomes narrative expedience	NEG
starts off witty and sophisticated and you want to love it – but filmmaker yvan attal quickly writes himself into a corner	NEG

Table 2: Some hard examples from movie review domain where a model fails at positive/negative sentiment classification.

others. Surface level patterns (based on word/tokens relations) are sparse. With linguistic annotations on text, we hope to discover more general patterns that capture the linguistic structure in text and are more frequent. By automatically extracting the subgraph features from the annotation graph, we hope to discover linguistic patterns commonly used in writing movie reviews that may not have been thought of earlier. Nonetheless, comparison with unigram only features is an important evaluation for subgraph features to justify the added complexity and extensive computation.

Automatically extracting subgraph features generate a large number of features especially when minimum support threshold ($minSup$) is low. As can be seen from Table 1, there are about 500,000 features for $minSup = 5$ (row 2) and about 7 million features for $minSup = 3$ (row 3). Our first strategy was to use all these features in an SVM model which is known to be robust to large feature spaces. As can be seen, subgraph features when added to the unigrams without any feature selection decrease the performance substantially (second and third rows in Table 1). This is possibly because of the increased feature space size with many redundant features among subgraphs.

Our next approach was to use popular feature selection approaches based on class association scores like χ^2 to select the most relevant features. We tried two versions of χ^2 based feature selection as described in Section 1, one based on significance and other based on fixed feature space size. The performance improves (compare rows 2 and 3 with rows 4,5 and 6) but it is still lower than unigram only features. We also tried other feature selection approaches discussed in Section 1. Feature Subsumption improves the performance substantially, but the performance is still much less than unigram only approach (compare row 7 with rows 1 and 2). Even correlation based filter (subgraph features with correlation ≥ 0.5 with their unigram features are filtered out) doesn’t help⁷.

With GP-based feature construction a few (30) most useful combination of features from the set of all unigram and subgraph features is created. When these GP feature combinations

⁷We only have results for $\chi^2 - size$, feature subsumption and correlation based feature selection for $minSup = 5$. We weren’t able run these feature selection approaches for $minSup = 3$ with the hardware we had as it required scoring and ranking 7 million features. But given that they didn’t improve performance for $minSup = 5$ (and other higher values of $minSup$ not reported here) much, we don’t expect the result to be very different for $minSup = 3$.

are added to the unigram features, we get a gain in performance over unigrams for both $minSup = 5$ (row 11) and $minSup = 3$ (row 13). The performance improves when we give more weight to precision than in previously published research (compare † and ‡). For $minSup = 3$ with more weight on precision (row 14), we observe a marginally significant gain ($p < 0.1$) in performance over unigrams alone, calculated using one-way ANOVA. The benefit from GP is more and significant only when subgraph features are used in addition to the unigram features for constructing more complex pattern features (compare rows 9-10 to 11-14 in Table 1).

A problem that we see with χ^2 feature selection is that several top-ranked features may be highly correlated. With GP-based feature construction, we can consider this relationship between features, and construct new features as a combination of selected unigram and subgraph features. With the correlation criterion in the evolution process, we are able to build combined features that provide new information compared to unigrams.

References

- Shilpa Arora, Elijah Mayfield, Carolyn Penstein Rosé, and Eric Nyberg. 2010. Sentiment classification using automatically extracted subgraph features. In *Proceedings of the Workshop on Emotion in Text at NAACL*.
- Mahesh Joshi and Carolyn Penstein Rosé. 2009. Generalizing dependency features for opinion mining. In *ACL-IJCNLP '09: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 313–316, Morristown, NJ, USA. Association for Computational Linguistics.
- Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA. Association for Computational Linguistics.
- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, July.
- Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. 2005. Sentiment classification using word sub-sequences and dependency sub-trees. In *Proceedings of PAKDD'05, the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*.
- Elijah Mayfield and Carolyn Penstein Rosé. 2010. Using feature construction to avoid large feature spaces in text classification. In *Proceedings of the Genetic and Evolutionary Computation Conference*.
- Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271–278.
- Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Morristown, NJ, USA. Association for Computational Linguistics.
- Ellen Riloff, Siddharth Patwardhan, and Janyce Wiebe. 2006. Feature subsumption for opinion analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 440–448. Association for Computational Linguistics.