Extending corpus-based identification of light verb constructions using a supervised learning framework

Yee Fan Tan, **Min-Yen Kan** and **Hang Cui**
Department of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543
{tanyeefa, kanmy, cuihang}@comp.nus.edu.sg

Abstract

Light verb constructions (LVCs) such as "make a call" and "give a presentation" pose challenges for natural language processing and understanding. We propose corpus-based methods to automatically identify LVCs. We extend existing corpus-based measures for identifying LVCs among verb-object pairs, using new features that use mutual information and assess the influence of other words in the context of a candidate verb-object pair, such as nouns and prepositions. To our knowledge, our work is the first to incorporate both existing and new LVC features into a unified machine learning approach. We experimentally demonstrate the superior performance of our framework and the effectiveness of the newly-proposed features.

1 Introduction

Many applications in natural language processing rely on the relationships between words in a discourse. Verbs play a central role in many such tasks; for example, the assignment of semantic roles to noun phrases in a sentence heavily depends on the verb that link the noun phrases together (as in "Pierre Vinken/SUBJ, will join/PRED, the board/OBJ").

However, verb processing is difficult because of many phenomena, such as nominalization of actions, verb particle constructions and light verb constructions. Applications that process verbs must handle these cases effectively. We focus on the identification of light verb constructions (also known as support verb constructions) in English, although such constructions play a prominent and productive role in many languages (Butt and Geuder, 2001; Miyamoto, 2000). Although the exact definition of an LVC varies in the literature, we use the following operational definition:

> A **light verb construction (LVC)** is a verb-complement pair in which the verb has little lexical meaning and much of the semantic content of the construction is obtained from the complement.

Examples of LVCs include "give a speech", "make good (on)" and "take (NP) into account". In the case in which the complement is a noun, it is often a deverbal noun and, as such, can usually be paraphrased using the object's root verb form without much loss in its semantics (*e.g.*, take a walk → walk, make a decision → decide, give a speech → speak).

We propose a corpus-based approach to determine whether a verb-object pair is a LVC. Note that we limit the scope of LVC detection to LVCs consisting of verbs with noun complements. Specifically, we extend previous work by examining how the local context of the candidate construction and the corpus-wide frequency of related words to the construction play an influence on the lightness of the verb.

A second contribution is to integrate our new features with previously reported ones under a machine

learning framework. This framework optimizes the weights for these measures automatically against a training corpus in supervised learning, and attests to the significant modeling improvements of our features on our corpus. Our corpus-based evaluation shows that the combination of previous work and our new features improves LVC detection significantly over previous work.

We first review previous corpus-based approaches to LVC detection, in Section 2. In Section 3, we show how we extend the use of mutual information and employ context modeling as features for improved LVC detection. We next describe our corpus processing and how we compiled our gold standard judgments used for supervised machine learning. Evaluation of several learning paradigms and feature combinations concludes the paper.

2 Related Work

With the recent availability of large corpora, statistical methods that use syntactic features are a current approach for many natural language tasks. This is the case for LVC detection as well.

Grefenstette and Teufel (1995) consider a similar task of identifying the most likely light verb for a given deverbal noun. Their approach focused on the deverbal noun and occurrences of the noun's verbal form in a corpus, arguing that the deverbal noun retains much of the verbal characteristics in the LVCs. To distinguish the LVC from other verb-object constructions, the deverbal noun must share similar argument/adjunct structures with its verbal counterpart. Verbs that appear often with these characteristic deverbal noun forms are deemed light verbs. They approximate the identification of argument/adjunct structures by using the preposition head of prepositional phrases that occur after the verb or object of interest.

Let n be a deverbal noun whose most likely light verb is to be found. Denote its verbal form by v', and let P be the set containing the three most frequently occurring prepositions that occur after v'. The verb-object pairs that are not followed by a preposition in P are then filtered out. For any verb v, let $g(v, n)$ be the count of verb-object pairs v-n that remain after the filtering step above. Grefenstette and Teufel proposed that the light verb for n be returned by the following equation:

$$\text{GT95}(n) = \arg\max_v g(v, n)$$

Interestingly, Grefenstette and Teufel indicated that their subsequent experiments suggested that the filtering step may not be necessary.

Whereas the GT95 measure centers on the deverbal object, Dras and Johnson (1996)'s measure considers also the verb's corpus frequency. The use of this symmetrical information improves LVC identification. Let $f(v, n)$ be the count of verb-object pairs occurring in the corpus, such that v is the verb, n is a deverbal noun. Then, the most likely light verb for n is given by:

$$\text{DJ96}(n) = \arg\max_v f(v, n) \sum_n f(v, n)$$

Stevenson *et al.* (2004)'s research examines evidence from constructions featuring determiners. They focused on expressions of the form v-a-n and v-det-n, where v is a light verb, n is a deverbal noun, a is an indefinite determiner (namely, "a" or "an"), and det is any determiner other than the indefinite. Examples of such constructions are "gave a speech", "take into account" and "take a walk". They employ mutual information which measures the frequency of co-occurrences of two variables, corrected for random agreement. We follow the literature and denote mutual information as $I(a, b)$. Then the following measure can be used:

$$\text{SFN04}(v, n) = 2 \times I(v, a\text{-}n) - I(v, det\text{-}n),$$

where higher values indicate a higher likelihood of v-a-n being a light verb construction. Also, they suggested that the determiner "the" be excluded from the development data since it frequently occurred in their data.

To summarize, LVC detection started by developing a single measure that utilized simple frequency counts of verbs and their complements. From this starting point, research has developed in two different directions: using more informed measures for word association (specifically, mutual information) and modeling how the context of the verb-complement pair.

Both the GT95 and DJ96 measures suffer from using frequency counts directly. Verbs that are not

light but occur very frequently (such as "buy" and "sell" in the Wall Street Journal) will be marked by these measures. They both *rank* rather than *decide* the possible light verbs. As such, given a deverbal noun, they sometimes suggest verbs that are not light. We hypothesize that substituting MI for frequency count can alleviate this problem.

The SFN04 metric adds in the context provided by determiners to augment LVC detection. This measure may work well for LVCs that are marked by determiners, but excludes a large portion of LVCs that are composed without determiners. To design a robust LVC detector requires integrating such specific contextual evidence with other general evidence.

3 Framework and Features

Previous work has shown that different measures based on corpus statistics can assist in LVC classification. However, it is not clear to what degree these different measures overlap or be used to reinforce each other's results. We aim to solve this problem by viewing LVC detection as a supervised classification problem. Such a framework can integrate the various measures and enable us to test their combinations in a generic manner. Specifically, each verb-object pair constitutes an individual classification instance, which possesses a set of features $f_1, \ldots, f_n$ and is assigned a class from $\{LVC, \neg LVC\}$. In such a machine learning framework, each of the aforementioned metrics are separate features. In our work, we have examined three different sets of features for LVC classification: (1) base, (2) extended and (3) new features.

We start by deriving three base features from key LVC detection measures as described by previous work – GT95, DJ96 and SFN04. As suggested in the previous section, we can make alternate formulations of the past work, such as to discard a pre-filtering step (*i.e.* filtering of constructions that do not include the top three most frequent prepositions). These measures make up the extended feature set. The final set of features are new and have not been used for LVC identification before. These include features that further model the influence of context (*e.g.* prepositions after the object) in LVC detection.

3.1 Base Features

Recall that the aim of the original GT95 and DJ96 formulae is to rank the possible support verbs given a deverbal noun. As each of these formulae contain a function which returns a numeric score inside the $\arg\max_v$ part, we use these functions as two of our base features derived from GT95 and DJ96:

$$\text{GT}(v, n) = g(v, n)$$

$$\text{DJ}(v, n) = f(v, n) \sum_n f(v, n)$$

On the other hand, the SFN04 measure can be used directly as our third base feature, and it will be referred to as SFN for the remaining of this paper.[1]

3.2 Extended Features

Since Grefenstette and Teufel indicated that the filtering step might not be necessary, *i.e.*, $f(v, n)$ may be used instead of $g(v, n)$, we also have the following extended feature:

$$\text{FREQ}(v, n) = f(v, n)$$

In addition, we experiment with the reverse process for the DJ feature, *i.e.*, to replace $f(v, n)$ in the function for DJ with $g(v, n)$, yielding the following extended feature:

$$\text{DJ-FILTER}(v, n) = g(v, n) \sum_n g(v, n)$$

In Grefenstette and Teufel's experiments, they used the top three prepositions for filtering. We further experiment with using all possible prepositions.

3.3 New Features

In our new feature set, we introduce features that we feel better model the v and n components as well as their joint occurrences v-n. We also introduce features that model the v-n pair's context, in terms of deverbal counts, prepositions and the number of noun complement modifiers, derived from our understanding of LVCs.

Most of these new features we propose are not good measures for LVC detection by themselves. It

[1] Recently, North (2005) has developed three additional metrics for LVC detection. We are looking at incorporating these metrics as well.

is because we use a machine learning framework, that enables us to leverage the small but noticeable influence that each of these features contribute to the final classification.

Mutual information

We observe that a verb v and a deverbal noun n are more likely to appear in verb-object pairs if they can form a LVC. To capture this evidence, we employ mutual information to measure the co-occurrences of a verb and a noun in verb-object pairs. Formally, the mutual information between a verb v and a deverbal noun n is defined as

$$I(v,n) = \log_2 \frac{P(v,n)}{P(v)P(n)},$$

where $P(v,n)$ denotes the probability of v and n constructing verb-object pairs. $P(v)$ is the probability of occurrence of v and $P(n)$ represents the probability of occurrence of n. Let $f(v,n)$ be the frequency of occurrence of the verb-object pair v-n and N be the number of all verb-object pairs in the corpus. We can estimate the above probabilities using their maximum likelihood estimates: $P(v,n) = \frac{f(v,n)}{N}$, $P(v) = \frac{\sum_n f(v,n)}{N}$ and $P(n) = \frac{\sum_v f(v,n)}{N}$.

However, $I(v,n)$ only measures the local information of co-occurrences between v and n. It cannot capture the global frequency of verb-object pair v-n, which is demonstrated as effective by Dras and Johnson (1996). As such, we need to combine the local mutual information with the global frequency of the verb-object pair. We thus have the following feature:

$$\text{MI-LOGFREQ} = I(v,n) \times \log_2 f(v,n)$$

Deverbal counts

Suppose a verb-object pair v-n is a LVC and the object n should be a deverbal noun. We denote v' to be the verbalized form of n. We thus expect that v-n should express the same semantic meaning as that of v'. However, verb-object pairs such as "have time" and "have right" are scored highly by the measures DJ and MI-LOGFREQ, even though the verbalized form of their objects, *i.e.*, "time" and "right", do not express the same meaning as the verb-object pairs do. Moreover, Grefenstette and Teufel's work have suggested that if the verb-object pair v-n is a light verb construction, n should share similar properties with v'. As such, we hypothesize that (1) the frequencies of n and v' should not differ very much, and (2) both frequencies are high given the fact that LVCs occur frequently in the text. Let's denote the frequencies of n and v' to be $f(n)$ and $f(v')$. We devise a novel feature based on the hypotheses:

$$\frac{\min(f(n), f(v'))}{\max(f(n), f(v'))} \times \min(f(n), f(v'))$$

where the two terms correspond to the above two hypotheses respectively.

Light verb classes

Since we extract the verb-object pairs from the Wall Street Journal section of the Penn Treebank, terms like "buy", "sell", "buy share" and "sell share" occur so frequently in the corpus that verb-object pairs like "buy share" and "sell share" are ranked high by most of the measures. However, "buy" and "sell" are not considered as light verbs. To overcome this problem, we predefine a list of possible light verbs. The predefined light verb list include "do", "get", "give", "have", "make", "put" and "take", which have been studied as light verbs in the literature. We thus define a feature that considers the verb in the verb-object pair: if the verb is in the predefined light verb list, the feature value is the verb itself; otherwise, the feature value is some other default value.

Noun sequence lengths

An object could be the last noun in a contiguous sequence of nouns, such as "list" in "priority watch list". We conjecture that a LVC is not likely to contain many continuous nouns because it is difficult to rewrite multiple nouns. To address this, we count the number of nouns in a contiguous sequence as a feature. Another reason of adopting this feature is that these are often proper nouns, such as "Greenville High School", which is not likely to contribute to a LVC.

Prepositions

We extract the preposition that heads the prepositional phase immediately after the object, if it exists. The idea stems from Grefenstette and Teufel's work, in which they use the prepositions to filter the verb-object pairs. Consider "we have the votes for this

candidate" versus "we have the votes in our hands". The former might be expressed as "we voted for this candidate" while the "vote" in the latter cannot be converted to a verb directly.

Other features

In addition to the above features, we also considered using the following features: the determiner before the object, the adjective before the object, and the number of words between the verb and its object. However these features did not improve performance significantly, so we omit them from further discussion.

4 Evaluation

In this section, we report the details of our experimental settings and results. First, we show how we constructed the labeled LVC corpus, which is employed as the gold standard in both training and testing using cross validation. Second, we describe the evaluation setup and discuss the experimental results obtained based on the labeled data.

4.1 Data Preparation

Some of the features we propose rely on the correct parsing of sentences. In order to minimize the errors incurred by incorrect parses, we employ the Wall Street Journal section in the Penn Treebank, which is manually parsed by linguists. We extract verb-object pairs from the Penn Treebank corpus and lemmatize them using WordNet's morphology module. As a filter, we require that a pair's object be a deverbal noun to be considered as a LVC. Specifically, we use WordNet to check whether a noun has a verb as one of its derivationally-related forms. A total of 24,647 candidate verb-object pairs are extracted, of which 15,707 are unique.

As the resulting dataset is too large for complete manual annotation given our resources, we sample the verb-object pairs from the extracted set. As most verb-object pairs are not LVCs, random sampling would not provide many positive LVC instances, and thus adversely affects the training of the classifier. Our aim in the sampling is to have balanced numbers of potential positive and negative instances. Based on the 24,647 verb-object pairs, we count the corpus frequencies of each verb v and each object n, denoted as $f(v)$ and $f(n)$. We also calculate the DJ-score of the verb-object pair DJ(v, n) by counting the pair frequencies. The data set is divided into 5 bins using $f(v)$ on a linear scale, 5 bins using $f(n)$ on a linear scale and 4 bins using DJ(v, n) on a logarithmic scale. We cross-multiply the three sets of bins and make them into $5 \times 5 \times 4 = 100$ bins. Finally, we uniformly sampled 2,840 verb-object pairs from all the bins to construct the data set for labeling.

4.2 Annotation

As noted by many linguistic studies, the verb in an LVC is often not completely vacuous, as they can serve to emphasize the proposition's aspect, its argument's semantics (*cf.*, θ roles), or other function (Butt and Geuder, 2001). As such, previous computational research had proposed that the "lightness" of an LVC might be best modeled as a continuum as opposed to a binary class (Stevenson et al., 2004).

We have thus annotated for two levels of lightness in our annotation of the verb-object pairs. Since the purpose of the work reported here is to flag all such constructions, we have simplified our task to a binary decision, similar to virtually all other previous, corpus-based work.

A website was set up for the annotation task, so that annotators can participate in it interactively. For each selected verb-object pair, a question is constructed by showing the sentence where the verb-object pair is extracted, as well as the verb-object pair itself. The annotator is then asked whether the presented verb-object pair is a LVC given the context of the sentence, and he or she will choose from the following options: (1) Yes, (2) Not sure, (3) No. The following three sentences illustrate the options.

(1) **Yes** – A Compaq Computer Corp. spokeswoman said that the company hasn't *made* a *decision* yet, although "it isn't under active consideration."

(2) **Not Sure** – Besides money, criminals have also used computers to steal secrets and intelligence, the newspaper said, but it *gave* no more *details*.

(3) **No** – But most companies are too afraid to *take* that *chance*.

The three authors, all natural language processing researchers, took part in our annotation task, and we asked all three of them to annotate on the same data. In total, we collected annotations for 741 questions. The average correlation coefficient between the three annotators is $r = 0.654$, which indicates fairly strong agreement between the annotators. We constructed the gold standard data by considering the median of the three annotations for each question. Two gold standard data sets are created:

- **Strict** – In the strict data set, a verb-object pair is considered to be a LVC if the median annotation is 1.
- **Lenient** – In the lenient data set, a verb-object pair is considered to be a LVC if the median annotation is either 1 or 2.

Each of the strict and lenient data sets have 741 verb-object pairs.

4.3 Experiment Setup

We have two aims for the experiments: (1) to compare the performance of classifiers using the base features and the extended features, and (2) to evaluate the effectiveness of our new features. Regarding the first aim, we make the following comparisons:

- GT (top 3 prepositions) versus GT (all prepositions) and FREQ
- DJ versus DJ-FILTER (top 3 prepositions and all prepositions)

In evaluations of our new features, we do not train the classifiers with only the new features, because each of these features alone does not make a measure of the lightness of verb-object pairs per se. As such, we evaluate these new features by adding them on top of the base features. We first construct a full feature set by utilizing the base features (DT, DJ and SFN) and all the new features. We chose not to add the extended features to the full feature set because these extended features are not independent to the base features. Next, to show the effectiveness of each new feature individually, we remove it from the full feature set and show the performance of classifiers without it.

We use the Weka data mining tool (Witten and Frank, 2000), which provides Java implementations of many classification algorithms. We employ four classifiers – NaïveBayes, KStar, DecisionTable and RandomTree, using different sets of features on both data sets. Stratified ten-fold cross-validation is performed. To facilitate comparison, for each feature set, we take the average of the F_1-measures (on the LVC class) obtained by all four classifiers to be the performance measure for that feature set.

4.4 Experimental Results

Table 1 shows the resulting F_1-measures of the classifiers using different sets of features in our experiments.

We make the following observations from Table 1:

1. The combinations of features outperform the individual features. We observe that using individual base features alone can achieve the highest average F_1-measure of 0.232 on the strict data set and 0.365 on the lenient data set respectively. When applying the combination of all base features, the average F_1-measures on both data sets increased significantly to 0.395 and 0.527. Previous work mainly studies individual statistics in identifying LVCs while ignoring the integration of various statistics. The results demonstrate that integrating different statistics (*i.e.* features) greatly boosts the performance of LVC identification. We attribute such improvements to that the base features are complementary to each other, and thus their integrated use is more comprehensive in characterizing LVCs in terms of global and local statistical evidence. More importantly, we employ off-the-shelf classifiers without special parameter tuning. This shows that generic machine learning methods can be applied to the problem of LVC detection. It provides a sound way to integrate various features to improve the overall performance.

   We also note that the features with filtering by prepositions achieve better performance than the corresponding features without filtering, except for DJ-FILTER (3 preps) on the lenient data set. The average increase in the F_1-measure by using preposition filtering is 0.068. Using all prepositions gives an even better per-

Feature(s)	Strict					Lenient				
	Naïve Bayes	KStar	Dec Table	Rnd Tree	Avg	Naïve Bayes	KStar	Dec Table	Rnd Tree	Avg
GT (3 preps)	0.188	0.000	0.237	0.208	0.158	0.150	0.000	0.493	0.328	0.243
GT (all preps)	0.333	0.000	0.300	0.293	0.232	0.185	0.000	0.425	0.303	0.229
FREQ	0.000	0.000	0.000	0.302	0.076	0.000	0.000	0.000	0.453	0.113
DJ	0.000	0.424	0.000	0.504	0.232	0.060	0.675	0.000	0.726	0.365
DJ-FILTER (3 preps)	0.270	0.289	0.000	0.424	0.246	0.204	0.226	0.431	0.476	0.334
DJ-FILTER (all preps)	0.340	0.364	0.000	0.460	0.291	0.242	0.294	0.496	0.509	0.385
SFN	0.000	0.000	0.030	0.000	0.008	0.000	0.000	0.272	0.000	0.068
BASE	0.218	0.515	0.368	0.479	0.395	0.161	0.669	0.604	0.674	0.527
+ MI-LOGFREQ	0.242	0.509	0.400	0.525	0.419	0.156	0.710	0.571	0.688	0.531
+ DEVERBAL	0.213	0.550	0.368	0.513	0.411	0.183	0.709	0.672	0.702	0.567
+ LV-CLASS	0.317	0.519	0.352	0.525	0.428	0.475	0.693	0.742	0.721	0.658
+ NOUNSEQ	0.218	0.509	0.382	0.471	0.395	0.164	0.678	0.604	0.656	0.525
+ PREP	0.255	0.533	0.315	0.464	0.392	0.172	0.721	0.596	0.650	0.535
+ NOUNSEQ + PREP	0.245	0.574	0.337	0.548	0.426	0.282	0.725	0.596	0.617	0.555
FULL	0.455	0.532	0.462	0.524	**0.493**	0.659	0.745	0.725	0.732	**0.715**
- MI-LOGFREQ	0.433	0.555	0.417	0.563	0.492	0.656	0.727	0.732	0.709	0.706
- DEVERBAL	0.411	0.508	0.466	0.571	0.489	0.634	0.747	0.732	0.717	0.707
- LV-CLASS	0.309	0.537	0.386	0.458	0.422	0.369	0.728	0.630	0.641	0.592
- NOUNSEQ	0.420	0.538	0.453	0.578	0.497	0.615	0.724	0.735	0.686	0.690
- PREP	0.364	0.532	0.457	0.544	0.474	0.641	0.730	0.725	0.753	0.712
- NOUNSEQ - PREP	0.265	0.550	0.438	0.556	0.452	0.638	0.717	0.735	0.727	0.704

Table 1: F_1-measures of classifiers for our evaluation.

formance than using only the top 3 prepositions in three out of the four cases. This suggests that preposition filtering can play an important role in LVC detection.

2. Our new features boost the overall performance. Applying the newly proposed features on top of the base feature set, *i.e.*, using the full feature set, gives average F_1-measures of 0.493 and 0.715 (shown in bold) in our experiments. We further show the effectiveness of each of the new features in two ways: to apply them individually on top of the base feature set, and to remove them individually from the full feature set. We find that when adding any of MI-LOGFREQ, deverbal counts or light verb classes to the base feature set, the F_1-measure is improved by 0.016-0.024 on the strict data set, and by 0.004-0.131 on the lenient data set. Conversely, when removing them from the full feature set, the average F_1-measures drop in a comparable scale. It shows that these new features boost the overall performance of the classifiers. We surmise that these new features are more task-specific and examine intrinsic features of LVCs. As such, integrated with the statistical base features, these features can be used to identify LVCs more accurately. It is worth noting that light verb class is a simple but important feature. It provides the highest average F_1-measure improvement compared to other new features. This is in accordance with the observation that different light verbs have different properties (Stevenson et al., 2004).

It is interesting to note that although noun sequence lengths and prepositions cannot individually bring much performance improvement, their combination augments the F_1-measure by 0.031 and 0.028 respectively on both data sets. We conjecture that these two features are less task-specific than other fea-

tures, and thus both features have to be combined with others to be effective.

5 Conclusions

Multiword expressions (MWEs) are probably one of the two main obstacles that hinder precise natural language processing (the other one is ambiguity) (Sag et al., 2002). As part of MWEs, LVCs remain least explored in the literature of computational linguistics. Past work addressed the problem of automatically detecting LVCs by employing single statistical measures. In this paper, we experiment on identifying LVCs using a uniform machine learning framework that integrates the use of various statistics. Moreover, we have extended the existing statistical measures and established new features to detect LVCs.

Our experimental results show that the integrated use of different features in a machine learning framework performs much better than using any of the features individually. In addition, we experimentally show that our newly-proposed features greatly boost the performance of classifiers that use base statistical features. Thus, our system achieves state-of-the-art performance over any previous approach for identifying LVCs. As such, we suggest that future work on automatic detection of LVCs employs a machine learning framework that combines complementary features, and examine intrinsic features that characterize the local context of LVCs, in order to achieve better performance.

A natural extension of this work is to apply such a machine learning framework to tackling the problem of detecting other kinds of MWEs automatically, for instance, verb particle constructions (VPCs) (Villavicencio, 2003). A VPC consists of a verb and one or more particles, such as "look up" and "leave out". Like LVCs, VPCs also show idiosyncrasies in their syntactic forms, which could be captured by a machine learning framework similar to the one proposed in this paper.

References

M. Butt and W. Geuder. 2001. On the (semi)lexical status of light verbs. In Norbert Corver and Henk van Riemsdijk, editors, *Semi-lexical Categories*, volume 59 of *Studies in Generative Grammar*, pages 323–370. Mouton de Gruyter.

M. Dras and M. Johnson. 1996. Death and lightness: Using a demographic model to find support verbs. In *5th International Conference on the Cognitive Science of Natural Language Processing*.

G. Grefenstette and S. Teufel. 1995. A corpus-based method for automatic identification of support verbs for nominalizations. In *Proceedings of EACL '95*.

T. Miyamoto. 2000. *The Light Verb Construction in Japanese. The role of the verbal noun.* John Benjamins, Amsterdam and Philadelphia.

R. North. 2005. Computational measures of the acceptability of light verb constructions. Master's thesis, University of Toronto.

I. Sag, T. Baldwin, F. Bond, A. Copestake, and D. Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Lecture Notes in Computer Science*, volume 2276, Jan, 2002.

S. Stevenson, A. Fazly, and R. North. 2004. Statistical measures of the semi-productivity of light verb constructions. In *2nd ACL Workshop on Multiword Expressions: Integrating Processing (MWE-2004)*, pages 1–8.

A. Villavicencio. 2003. Verb-particle constructions in the world wide web. In *Proceedings of the ACL-SIGSEM Workshop on the Linguistic Dimensions of Prepositions and their use in Computational Linguistics Formalisms and Applications*.

I. Witten and E. Frank. 2000. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco.