Learning Count Classifier Preferences of Malay Nouns

Jeremy Nicholson and **Timothy Baldwin**
NICTA Victoria Research Laboratories
University of Melbourne, VIC 3010, Australia
{jeremymn,tim}@csse.unimelb.edu.au

Abstract

We develop a data set of Malay lexemes labelled with count classifiers, that are attested in raw or lemmatised corpora. A maximum entropy classifier based on simple, language-inspecific features generated from context tokens achieves about 50% F-score, or about 65% precision when a suite of binary classifiers is built to aid multi-class prediction of headword nouns. Surprisingly, numeric features are not observed to aid classification. This system represents a useful step for semi-supervised lexicography across a range of languages.

1 Introduction

This work examines deep lexical acquisition (DLA) of count classifiers in Malay.[1] DLA is the process of (semi-)automatically learning linguistic structures for use in linguistically-rich language resources such as precision grammars or wordnets (Baldwin, 2007). Malay is a significant target for DLA in that relatively few NLP resources have been developed for it.[2]

In many languages — notably many South-East and East Asian languages — numbers cannot generally modify nouns. Instead, they modify count classifiers[3] which tend to occur as a genitive modifier or in apposition to the noun. In (1) below, *biji* is an example of a Malay count classifier (CL), premodified by the numeral *tiga* "three"; similarly for the Japanese in (2):[4]

(1) *tiga* *biji* *pisang*
three CL banana
"three bananas"

(2) *saN* *boN* *no* *banana*
three CL GEN banana
"three bananas"

The closest English analogue to count classifiers is measure nouns (Flickinger and Bond, 2003), like *loaf* in *3 loaves of bread* (compare *3 breads*, which has markedly different semantics). Measure nouns cannot be used to count countable nouns in English, however (e.g. ∗*3 loaves of roll*).

Syntactically, count classifiers are nouns which occur with a (usually numeric) specifier, and are used to quantify uncountable nouns. In languages such as Malay, most nouns are uncountable and numeral–classifier combinations must be used to enumerate instances of a given noun (c.f. ∗*tiga pisang*; this is not true of the nouns that are classifiers themselves, like *orang* "person" or *buah* "fruit", e.g. *tiga orang* "three people"). Semantically, count classifiers select for objects of particular semantics, commonly determined by a conceptual class (e.g. HUMAN or BIRD) or the relative dimensions of the object (e.g. LONG-AND-THIN or FLAT).

[1] The language is called by a number of names, including bahasa Malayu, bahasa Malaysia, the Malaysian language, and Standard Malay. We simply use *Malay*, to agree with the Ethnologue (Gordon, 2005).

[2] Although some research has been done on bahasa Indonesia (henceforth *Indonesian*), another Malayan language with which Malay is (somewhat) mutually intelligible.

[3] This paper focuses specifically on sortal classifiers. See Paik and Bond (2001) for discussion of other types of count classifiers, namely event, mensural, group and taxonomic.

[4] In the Japanese example, the genitive (GEN) linker is required to connect the numeral–classifier combination with the noun in a single noun phrase.

In the case of Malay, e.g., *biji* is the count classifier for FRUIT, while *orang* is the count classifier for HUMAN:

(3) *empat orang raja*
four CL king
"four kings"

(4) *#empat biji raja*
four CL king
"four kings" (intended)

There are over 50 such count classifiers in Malay, each with differing semantic restrictions.

Count classifiers are highly significant in languages such as Malay when generating text, because a noun specified with an incorrect classifier can lead to unintended or infelicitous readings (c.f. *#empat biji raja* above), or highly marked language. Other languages, like Thai, require classifiers not only for specification but also stative verb modification. Additionally, the choice of classifier can help disambiguate noun sense, either for unknown nouns or for known ambiguous nouns. In order to generate correctly or use count classifiers as a source of disambiguation, however, we require a large-coverage noun lexicon with information on classifier compatibility. The objective of this paper is to automatically generate such a lexicon for Malay. As far as we are aware, this is the first such attempt for Malay.

The proposed approach to learning the classifier preferences for Malay nouns is to represent nouns via their distributional context, i.e. the tokens they commonly co-occur with in corpus data within a fixed context window. We model distributional context in a standard supervised learning framework, generalising from training examples to classify novel nouns. In this, we experiment with different distributional representations (words vs. lemmas, and different n-gram orders), and compare our proposed system to a system based on direct analysis of noun co-occurrence with different numeral–classifier combinations.

This research forms part of a broader project on the general applicability of DLA across a range of word learning tasks. As such, we propose a model that is as language-inspecific as possible, making only the bare minimum assumptions about Malay such as the fact that a modifier tends to occur within a few tokens of its head. As a result, we expect that the observed results will scale to other languages with count classifier systems such as Japanese, Chinese, or Thai. The major difference in porting the method to other languages would be access to various lexical resources — for example, lexical acquisition would be quite difficult for these three languages without a tokeniser — where available parsers or ontologies might capture syntactic or semantic similarities much more easily than surface cues.

As far as we are aware, this paper presents the first ever results for count classifier-based classification of Malay nouns.

2 Background

2.1 NLP for Malay

As the basis of our evaluation, we use the KAMI Malay–English translation dictionary (Quah et al., 2001). Although primarily a translation dictionary, with translations of Malay headwords into both English and Chinese, KAMI contains syntactic information and semantics in terms of a large ontology. Of the total of around 90K lexical entries in KAMI, we make use of around 19K nominal lexical entries which are annotated for headword, lemma and POS tag when specified, and count classifier.

As a corpus of Malay text, we use the 1.2M-token web document collection described in Baldwin and Awab (2006).[5] The corpus has been sentence- and word-tokenised, and additionally lemmatised based on a hand-crafted set of inflectional and derivational morphology rules for Malay, developed using KAMI and an electronic version of a standard Malay dictionary (Taharin, 1996).

While little NLP has been performed for Malay, Indonesian has seen some interesting work in the past few years. Adriani et al. (2007) examined stemming Indonesian, somewhat overlapping with Baldwin and Awab above, evaluating on the information retrieval testbed from Asian et al. (2004). Recently, a probabilistic parser of Indonesian has been developed, as discussed in Gusmita and Manu-

[5] An alternative corpus of Malay would have been the Dewan Bahasa & Pustaka Corpus, with about 114M word tokens. As it is not readily accessible, however, we were unable to use it in this research.

rung (2008), and used for information extraction and question answering (Larasati and Manurung, 2007).

2.2 Count Classifiers

Shirai et al. (2008) discuss the theory and difficulties associated with automatically developing a broad taxonomy of count classifiers suitable for three languages: Chinese, Japanese, and Thai. They find that relatively high agreement between Chinese and Japanese, but Thai remains resistant to a universal hierarchy of count classes.

Otherwise, work on count classifiers has mostly focussed on a single language at a time, primarily Japanese. Bond and Paik (1997) consider the lexical implications of the typology of kind and shape classifiers, and propose a hierarchy for Japanese classifiers that they extend to Korean classifiers. Bond and Paik (2000) examine using the semantic class of a Japanese noun from an ontology to predict its count classifier, and Paik and Bond (2001) extend this strategy to include both Japanese and Korean. Finally, Sornlertlamvanich et al. (1994) propose their own typology for classifiers within Thai, and an automatic method to predict these using corpus evidence.

As for comparable structures in English, Flickinger and Bond (2003) analyse measure nouns within a Head-Driven Phrase Structure Grammar framework. Countability, which is a salient feature of nouns in English, undergoes many of the same structural syntax and semantic preferences as count classifiers. Baldwin and Bond (2003) motivate and examine a variety of surface cues in English that can be used to predict typewise countability of a given noun. Taking a mapped cross-lingual ontology, van der Beek and Baldwin (2004) use the relatedness of Dutch and English to cross-lingually predict countability, and observe comparable performance to monolingual prediction.

3 Methodology

3.1 Data Set

First, we constructed a data set based on the resources available. As we chose to approach the task in a supervised learning framework, we required a set of labelled exemplars in order to train and test our classifiers.

To obtain this, we first extracted the approximately 19K noun entries from KAMI (Quah et al., 2001) for which at least one count classifier was attested. For each of these, we recorded a wordform and a lemma. The wordform was extracted directly from the lexical entry, and included both simplex words and multiword expressions. The lemma also came directly from the lexical entry where a lemma was provided, and in instances where no lemma was found in KAMI, we used a case-folded version of the headword as the lemma. We looked up each noun in the corpus based on the wordform in the original text data, and also the lemma in the lemmatised version of the corpus.

The gold-standard data set was generated from those lexical entries for which we were able to find at least one instance of the wordform in the raw corpus or at least one instance of the lemma in the lemmatised corpus. This resulted in a total of 3935 unique nouns. The total number of number classifiers attested across all the nouns was 51.

We also generated three reduced data sets, by: (1) excluding proper nouns, based either on the POS tag in KAMI or in the case that no POS was found, the capitalisation of the headword (3767 unique nouns); (2) excluding multiword expressions (MWEs: Sag et al. (2002)), as defined by the occurrence of whitespace in the headword (e.g. *abang ipar* "brother-in-law"; 2938 unique nouns); and (3) excluding both proper nouns and MWEs (2795 unique nouns). The underlying assumption was that proper nouns and MWEs tend to have obfuscatory syntax or semantics which could make the task more difficult; it would seem that proper nouns and MWEs should be handled using a dedicated mechanism (e.g. choosing the head of a compound noun from which to select a classifier).

3.2 Features

For each exemplar within the gold-standard data set, we developed a feature vector based on instances from the corpus. To limit the language specificity of the feature set, we built features by considering a context window of four tokens to the left and right for each corpus instance. The rationale for this is that while numerous languages permit free global word order, local word order (say, of an NP) tends to be more rigid. Windows of sizes less than four

tended to reduce performance slightly; larger windows were not observed to significantly change performance.

For each instance of the wordform (i.e. the headword or headwords of a lexical entry from KAMI) in the raw corpus, we generated a feature vector of each of the (up to) four preceding and (up to) four following wordform tokens within the boundaries of the sentence the wordform occurred in. We additionally index each context wordform by its relative position to the target word. The same was done with the lemma of the target word, using the lemmatised corpus instead of the raw corpus. Consequently, there were numerous mismatches where a context window appeared in the wordform features and not the lemma features (e.g. if a morphologically-derived wordform was not listed in KAMI with its lemma), or *vice versa*. The combined feature vector for a given noun instance was formed by concatenating all of the context window features, as well as the target word wordform and lemma.

One possible extension that we chose not to follow on the grounds of language specificity is the use of morphological features. Malay has a number of prefixes, suffixes, and circumfixes which could provide information for selecting a classifiers. Specifically, the agentive prefix *me-* could be indicative of the *orang* class.

3.3 Classifiers

To build our classifiers, we used a maximum-entropy learner.[6] It is not uncommon for a noun to be listed with multiple classifiers in KAMI, such that we need a classifier architecture which supports multi-class classification. We experimented with two modes of classification: a monolithic classifier, and a suite of binary classifiers.

For the monolithic classifier, we adopt the equivalent of a first-sense classifier in word sense disambiguation terms, that is we label each training instance with only the first count classifier listed in KAMI. The underlying assumption here is that the first-listed count classifier is the default for that noun, and thus likely to have the best fit with the distributional model of that noun. As such, out of all the possible count classifiers, we expect to have the best chance of correctly predicting the first-listed classifier, and also expect it to provide the best-quality training data. When we restrict the set of annotations to only the first-listed classifiers, we reduce the set of possible classes to 42, out of which 10 are singletons and therefore never going to be correctly classified. In evaluating the monolithic classifier, we calculate precision and recall across the full set of actual count classifiers listed in KAMI (i.e. not just the first-listed count classifier).

For the suite of binary classifiers, we construct a single classifier for each of the 51 classes (of which 16 are singleton classes, and hence disregarded). For a given classifier, we categorised each exemplar according to whether the particular class is an acceptable count classifier for the headword according to KAMI. In testing, the classifier posits an affirmative class assignment when the posterior probability of the exemplar being a member of that class is greater than that of it not being a member. This mode of classification allows multi-classification more easily than that of the monolithic classifier, where multi-classes are too sparse to be meaningful. Evaluation of the suite is in terms of precision and recall averaged across the entire set of classifiers.

When selecting the training and test splits, we use 10-fold stratified cross-validation. Briefly, we separate the entire data set into 10 distinct partitions which each have generally the same class distribution as the entire data set. Each of the 10 partitions is used as a test set, with the other 9 as training partitions, and the results are micro-averaged across the 10 test sets. This evaluation strategy is used for both the monolithic classifier and the binary suite. 10-fold stratified cross-validation has been variously shown to tend to have less bias and variance than other hold-out methods, thereby giving a better indication of performance on unseen data.

4 Results

4.1 Comparison of Feature Sets

First, we examined the use of wordform features, lemma features and the combination of the two. We contrast the precision (P), recall (R), and F-score $F_{\beta=1} = \frac{2PR}{P+R}$ for the monolithic classifier (Mono) and the suite of binary classifiers (Suite) across the

[6] We used the OpenNLP implementation available at `http://www.sourceforge.net/projects/maxent/`.

				Mono			Suite		
Token	LEs	n-gram	Numb	P	R	F	P	R	F
Baseline	All			43.8	41.5	42.6	43.8	41.5	42.6
Baseline	−PN −MWE			42.4	39.3	40.8	42.4	39.3	40.8
W	All	1	No	48.3	45.2	46.7	76.4	33.7	46.8
L	All	1	No	49.7	47.0	48.3	61.0	38.3	47.1
W+L	All	1	No	57.6	54.7	56.1	69.3	42.3	52.5
	−PN	1	No	55.7	52.6	54.1	71.4	41.5	52.5
	−MWE	1	No	53.3	49.6	51.4	64.3	40.6	49.8
	−PN −MWE	1	No	51.8	47.9	49.8	65.6	38.3	48.4
		1+2	No	50.2	45.8	47.9	63.2	37.4	47.0
		1	Yes	52.1	48.3	50.1	65.7	38.2	48.3
		1+2	Yes	50.3	45.9	48.0	63.2	37.4	47.0

Table 1: Evaluation (in %) of the monolithic classifier (Mono) and suite of binary classifiers (Suite) in terms of precision (P), recall (R), and F-score (F). The majority class baseline is also presented. (The Token set is one of wordform feature set (W), lemma feature set (L), and the combined set (W+L); the lexical entries (LEs) under consideration were the entire set (All), or all except proper nouns (−PN), or all except MWEs (−MWE), or all except proper nouns or MWEs (−PN −MWE); the n-grams are either unigrams (1) or mixed unigrams and bigrams (1+2); and a given model optionally makes use of number folding (Numb)).

three feature sets: wordform features only (W), lemma features only (L), and both wordform and lemma features (W+L). The results are summarised under the Token heading in Table 1.

The F-score of the monolithic classifier and suite of binary classifiers increases from wordform features, to lemma features, to both. One reason for this is that there are simply more lemma features than wordform features (140 to 30 windows on average, for a given instance). This is particularly evident when we consider that the lemma features have higher recall but lower precision than the wordform features over the suite. We find that the wordform features are generally more accurate, allowing the classifier to tend to classify fewer instances with greater confidence, while noise gets introduced to the lemma feature classifier caused by folding derivational morphology in the lemmatised corpus. For example, features for *kemudi* "to drive" and *mengemudi* "driving" will be erroneously generated when considering *pengemudi* "driver" because they share a lemma, and verbal context usually does not reliably indicate the count classifier of the noun.

Despite the overgeneration caused by the lemma features, the combination of the wordform and lemma features leads to the classifier with the best F-score. The same holds true for our later experiments, so we only report results on the combination of feature sets below.

In addition to providing direct insight into the optimal feature representation for count classifier learning, these results represent the first extrinsic evaluation of the Baldwin and Awab Malay lemmatiser, and suggest that lemmatiser is working effectively.

4.2 Comparison of Data Sets

Next, we contrasted excluding the proper noun and MWE lexical entries from KAMI. We generated four data sets: with both proper noun and MWE entries (All), with proper nouns but without MWE (−MWE), without proper nouns but with MWE entries (−PN), and without proper noun or MWE entries (−PN −MWE). The precision, recall, and F-score for wordform and lemma features are shown in Table 1, under the LEs heading.

Removing either proper noun or MWE entries caused the F-score to drop. The effects are somewhat more subtle when considering the precision–recall trade-off for the suite of classifiers.

Excluding proper noun entries from the data set causes the precision of the suite of classifiers to rise, but the recall to fall. This indicates that the removed entries could be classified with high recall but low

precision. Examining the proper noun entries from KAMI, we notice that most of them tend to be of the class *orang* (the person counter, as other semantic classes of proper noun do not tend to be counted: consider *?four New Zealands* in English) — this is the majority class for the entire data set. Removing majority class items tend to lower the baseline and make the task more difficult.

Excluding MWE entries from the data set causes both the precision and the recall of the suite to drop. This indicates that the removed entries could be classified with high precision and high recall, suggesting that they are generally compositional (e.g. *lap dog* would take the semantics of its head *dog* and is hence compositional, while *hot dog* is not). In KAMI, MWEs are often listed with the lemma of the semantic head; if the simplex head is an entry in the training data, then it becomes trivial to classify compositional MWEs without considering the noisy context features.

So, contrary to our intuition, proper nouns and MWEs tend to be easier to classify than the rest of the data set as a whole. This may not be true for other tasks, or even the general case of this task on unseen data, but the way we generate the data set from KAMI could tend to overestimate our results from these classes of lexeme. Consequently, we only describe results on the data set without proper nouns or MWEs below, to give a more conservative estimate of system performance.

4.3 Monolithic vs. Suite Classification

We consider the baseline for the task to be that of the majority class: *orang*. It corresponds to 1185 out of 2795 total unique headwords in the restricted data set from KAMI: baseline precision is 42.4%, recall is 39.3% when multiple headword–class assignments are considered. In Table 1 we present both the baseline performance over all lexical entries, and that over all lexical entries other than proper nouns and MWEs. Note that the baseline is independent of the classifier architecture (monolithic or binary suite), but duplicated across the two columns for ease of numerical comparison.

Both the monolithic classifier and suite significantly outperform the baseline in terms of F-score, the monolithic somewhat more so ($\chi^2 < 0.1, \mathrm{P} < 1$). Notably, *orang* entries made up about 40% of the data set, and the generic (i.e. the most semantically inspecific) count classifier *buah* made up about another 30% of the lexemes. It seemed that, since the 70% of the dataset covered by these two classes was greater than our performance, a classifier which could reliably distinguish between these two classes — while ignoring the other 49 classes — could have commensurately higher performance. (In effect, this would require distinguishing between persons for *orang* and things for *buah*, semantically speaking.) While the performance of that two-class system was greater for those classes under consideration, overall performance was slightly worse.

The suite of classifiers generally had higher precision and lower recall than the monolithic classifier. This indicates that the suite tended to make fewer positive assignments, but was more accurate when it did so. While the monolithic classifier was forced to make a classifier decision for every instance, the suite of classifiers tended to be conservative in classification, and consequently ended up with numerous headwords for which it did not predict a count classifier at all. This is particularly unsurprising when we consider that the prior probability for every category was less than 50%, so the default maximum likelihood is to define no headword as belonging to a given class.

4.4 Number features

As a further experiment, we examined the set of context windows where one of the tokens was numeric. It seems reasonable for a lexicographer to have defined the extent of numeric tokens ahead of time, given a moderately large corpus; numerals are of particular importance to the selection of a count classifier, in that they indicate the preference of how a given countable noun is counted.

We list the possible word-based numeral tokens below, which we combine with a numeric regular expression. Compound number formation is similar to English, and tokens beneath *sembilan* "nine" in the table are usually prefixed by *se-* as a replacement for a *satu* premodifier. (For example, *sebelas* instead of *satu belas* for "eleven".)

We attempted to classify after folding all tokens of this nature into a single "numeric" feature (Numb in Table 1), thereby allowing different numbers to be directly compared. The net effect was to raise

Token	Description
satu	"one"
dua	"two"
tiga	"three"
empat	"four"
lima	"five"
enam	"six"
tujuh	"seven"
lapan	"eight"
sembilan	"nine"
belas	"-teen"
puluh	"ten"
ratus	"hundred"
ribu	"thousand"
juta	"million"
bilion	"billion"
trilion	"trillion"

Table 2: Numbers in Malay, with English glosses.

the F-score of the monolithic classifier from 49.8% to 50.1%, and to lower the F-score of the suite of classifiers from 48.4% to 48.3%. Neither of these results are significantly different.

We also considered using a strategy more akin to what a human might go through in attempting to categorise nouns, in looking for all direct attestations of a given noun in the immediate context of a number expression, excluding all other lexical contexts (ideally being left with only contexts of the type "number CL noun", like *tiga biji pisang* above). However, the net effect was to remove most context windows, leaving numerous exemplars which could not be classified at all. Predictably, this led to an increase in precision but large drop in recall, the net effect of which was a drop in F-score. Folding numeric features here only improved the F-score by a few tenths of a percent.

Folding the numeric features turned out not to be valuable, as the fact that a given noun has numerals in its context is not a strong indicator of a certain type of count classifier, and all training examples were given an approximately equal benefit. Instead, we considered bi-word features, hoping to capture contexts of the type *tiga biji* "three CL", which seem to be a better indication of count classifier preference. A comparison of uni-word features and the combination of uni-word and bi-word features is shown in Table 1 under the n-gram heading.

The net effect of adding bi-word features is to reduce performance by one or two percent. One possible reason for this is the addition of many features: a context half-window of four tokens generates six more bi-word features.

Once more, excluding context windows without a numeral expression feature causes performance to drop drastically to 32.6% and 37.6% F-score for the monolithic classifier and suite, respectively. Folding numeric expression features again only changes figures by about one-tenth of one percent.

5 Discussion

We achieved about 50% F-score when attempting to predict the count classifiers of simplex common nouns in Malay. Although getting every second prediction incorrect seems untenable for automatic creation of a reliable lexical resource, we can see a system such as this being useful as a part of a semi-automatic system, for example, by providing high-confidence predictions of missing lexical entries to be judged by a human lexicographer. The precision of the suite of classifiers seems promising for this, having an error rate of about one in three predictions.

To our knowledge, there are no studies of Malay count classifiers to which we can directly compare these results. One possibility is to extend the work to Indonesian count classifiers — the mutual intelligibility and similar syntax seem amenable to this. However, it seems that there is a systematic divergence between count classifiers in these two languages: they are optional in many cases in Indonesian where they are not in Malay, and some counters do not map across well (e.g. *bilah*, the counter for knives and other long, thin objects, is not a count classifier in Indonesian). On the other hand, our approach does not rely on any features of Malay, and we expect that it would pick up surface cues equally well in Indonesian to achieve similar performance. The use of parsers like that of Gusmita and Manurung (2008) could provide further benefits.

When examining the feature sets we used, we

discovered that features generated on the wordform level tended to be higher precision than features generated from the lemmatised corpus. One reason for this is that regular inflectional or derivational morphology is stripped to give numerous misleading contexts in the corpus. Consequently, we expect that some entries in the data set are added erroneously: where the wordform is not attested in the raw corpus and all of the instances of the lemma in the lemmatised corpus correspond to verbal or other derived forms. Given that almost a third of the data, or 945 lexemes, had no wordform instances, we might be underestimating our recall to some extent. With the aid of a POS tagger, we could condition the lemma windows in the feature set on a nominal POS for the candidate, and we would expect precision for these features to increase.

We observed that proper nouns and MWEs were less difficult to classify than the data set as a whole. Countable proper nouns tend to be people (e.g. *Australians* in English), and take *orang* as a counter, so for a given proper noun, we expect that the countable–uncountable cline to be the only hurdle to choosing the appropriate count classifier. MWEs tend to be compositional and hence share their class with their head, and an independent strategy for determining the head of the expression could transform such an entry into an easier simplex one.

The monolithic classifier, to choose between the entire set of 35 or so classes (after removing singleton classes), had a better F-score than the suite of binary classifiers, which predicted the suitability of a given count classifier one class at a time. However, the suite tended to be more conservative in its classifications, and had greater precision than the monolithic classifier. While high recall is useful in many circumstances, we feel that precision is more important as part of a pipeline with a lexicographer making gold-standard judgements. This trade-off makes combination of the two systems seem promising. One way could be to classify a given headword using both systems, and vote for the preferred class. Confidence of the classification, as given by the posterior probability of the maximum entropy model, could be taken into account. A two-tiered system could also be used, where the suite could be used to predict a high-precision class, and then the monolithic classifier could force a prediction for the entries where no classes at all were assigned by the suite. That case, where no classes are assigned, is particularly problematic with generation, for example, and other circumstances when a decision is required.[7]

Numeric features, we discovered, were only theoretically useful for bi-word contexts. Even then, we discerned little benefit when tailoring the feature vector to them. One possibility was that there was too much noise amongst the bigram features for the classifier to reliably use them as context. We feel, however, that restricting oneself to the syntax of numeral count expressions oversimplifies the problem somewhat. Since the preferred count classifier for a given noun tends to describe some underlying semantic property, the distributional hypothesis contends that surface cues from all contexts provide evidence. Our data agrees with this to some extent.

6 Conclusion

We developed a data set of labelled lexemes in Malay corresponding to headwords and count classifiers which were attested in a raw or lemmatised corpus. A maximum entropy classifier based on features generated from context tokens achieved about 50% F-score in prediction, and about 65% precision when a suite of binary classifiers was used. We envisage such a system to be a useful part of semi-automatic lexicography over a range of languages.

Acknowledgments

We would like to thank Francis Bond for his wisdom and advice throughout this research. NICTA is funded by the Australian government as represented by Department of Broadband, Communication and Digital Economy, and the Australian Research Council through the ICT Centre of Excellence programme.

References

Mirna Adriani, Jelita Asian, Bobby Nazief, Seyed M. M. Tahaghoghi, and Hugh E. Williams. 2007. Stemming Indonesian: A confix-stripping approach. *ACM Transactions on Asian Language Information Processing*, 6:1–33.

Jelita Asian, Hugh E. Williams, and Seyed M. M. Tahaghoghi. 2004. A testbed for Indonesian text

[7] The issue of no classes being assigned is interesting when the countability–uncountability cline is considered. Potentially this could be an indication that the noun is, in fact, weakly countable. See, for example, Bond et al. (1994).

retrieval. In *Proceedings of the 9th Australasian Document Computing Symposium*, pages 55–58, Melbourne, Australia.

Timothy Baldwin and Su'ad Awab. 2006. Open source corpus analysis tools for Malay. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 2212–5, Genoa, Italy.

Timothy Baldwin and Francis Bond. 2003. Learning the countability of English nouns from corpus data. In *Proceedings of the 41st Annual Meeting of the ACL*, pages 463–470, Sapporo, Japan.

Timothy Baldwin. 2007. Scalable deep linguistic processing: Mind the lexical gap. In *Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation*, pages 3–12, Seoul, Korea.

Francis Bond and Kyonghee Paik. 1997. Classifying correspondence in Japanese and Korean. In *Proceedings of the 3rd Conference of the Pacific Association for Computational Linguistics*, pages 58–67, Tokyo, Japan.

Francis Bond and Kyonghee Paik. 2000. Reusing an ontology to generate numeral classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 90–96, Saarbrücken, Germany.

Francis Bond, Kentaro Ogura, and Satoru Ikehara. 1994. Countability and number in Japanese-to-English machine translation. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 32–38, Kyoto, Japan.

Dan Flickinger and Francis Bond. 2003. A two–rule analysis of measure noun phrases. In *Proceedings of the 10th International Conference on Head–Driven Phrase Structure Grammar*, pages 111–121, East Lansing, USA.

Raymund G. Gordon, Jr, editor. 2005. *Ethnologue: Languages of the World, Fifteenth Edition*. SIL International.

Ria Hari Gusmita and Ruli Manurung. 2008. Some initial experiments with Indonesian probabilistic parsing. In *Proceedings of the 2nd International MALINDO Workshop*, Cyberjaya, Malaysia.

Septina Dian Larasati and Ruli Manurung. 2007. Towards a semantic analysis of bahasa Indonesia for question answering. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 273–280, Melbourne, Australia.

Kyonghee Paik and Francis Bond. 2001. Multilingual generation of numeral classifiers using a common ontology. In *Proceedings of the 19th International Conference on the Computer Processing of Oriental Languages*, pages 141–147, Seoul, Korea.

Chiew Kin Quah, Francis Bond, and Takefumi Yamazaki. 2001. Design and construction of a machine-tractable Malay-English lexicon. In *Proceedings of the 2nd Biennial Conference of ASIALEX*, pages 200–205, Seoul, Korea.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Proc. of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2002)*, pages 1–15, Mexico City, Mexico.

Kiyoaki Shirai, Takenobu Tokunaga, Chu-Ren Huang, Shu-Kai Hsieh, Tzu-Yi Kuo, Virach Sornlertlamvanich, and Thatsanee Charoenporn. 2008. Constructing taxonomy of numerative classifiers for Asian languages. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, pages 397–402, Hyderabad, India.

Virach Sornlertlamvanich, Wantanee Pantachat, and Surapant Meknavin. 1994. Classifier assignment by corpus-based approach. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 556–561, Kyoto, Japan.

Mashitah Taharin, editor. 1996. *Kamus Dewan Edisi Ketiga*. Dewan Bahasa Dan Pustaka, Kuala Lumpur, Malaysia.

Leonoor van der Beek and Timothy Baldwin. 2004. Crosslingual countability classification with EuroWordNet. In *Papers from the 14th Meeting of Computational Linguistics in the Netherlands*, pages 141–155, Antwerp, Belgium.