Evaluation of Classification Algorithms and Features for Collocation Extraction in Croatian

Mladen Karan, Jan Šnajder, Bojana Dalbelo Bašić

University of Zagreb
Faculty of Electrical Engineering and Computing

mladen.karan@fer.hr, jan.snajder@fer.hr, bojana.dalbelo@fer.hr

Abstract

Collocations can be defined as words that occur together significantly more often than it would be expected by chance. Many natural language processing applications such as natural language generation, word sense disambiguation and machine translation can benefit from having access to information about collocated words. We approach collocation extraction as a classification problem where the task is to classify a given n-gram as either a collocation (positive) or a non-collocation (negative). Among the features used are word frequencies, classical association measures (Dice, PMI, chi2), and POS tags. In addition, semantic word relatedness modeled by latent semantic analysis is also included. We apply wrapper feature subset selection to determine the best set of features. Performance of various classification algorithms is tested. Experiments are conducted on a manually annotated set of bigrams and trigrams sampled from a Croatian newspaper corpus. Best results obtained are 79.8 F1 measure for bigrams and 67.5 F1 measure for trigrams. The best classifier for bigrams was SVM, while for trigrams the decision tree gave the best performance. Features which contributed the most to overall performance were PMI, semantic relatedness, and POS information.

Keywords: collocation extraction, feature subset selection, Croatian language

1. Introduction

Automatic collocation extraction (CE) is the task of automatically identifying collocated words in a given natural language text. The term *collocation* has a significant overlap with the term *multi word entity* (MWE). MWEs include phrases, idioms, named entities, etc. Collocations can be viewed as empirical epiphenomena of MWEs: each time a MWE is mentioned in a text, the words forming it occur together. Most collocations have a certain degree of added meaning, making them more than a sum of their parts. While there exist more elaborate definitions, in the scope of this paper we will define collocations as sequences of terms or words that appear together more often than it would be expected by chance (Manning and Schütze, 1999).

The reason why CE is important is that many Natural Language processing (NLP) tasks can benefit by having access to information about collocated words. One example of a task that greatly benefits from such information is natural language generation (NLG) in the form of text or speech. A common example is the phrase *"strong tea"* used far more often than *"powerful tea"*, which sounds unnatural, although it is grammatically correct and conveys the same meaning. This information is very useful to an NLG algorithm. Some other areas of NLP that benefit from collocation information include word sense disambiguation (Jimeno-Yepes et al., 2011; Jin et al., 2010) as well as machine translation (Liu et al., 2010).

CE can be framed as a classification problem, where candidates are classified as collocations or non-collocations based on input features. Traditionally used lexical association measures (AMs) used for CE (Church and Hanks, 1990) have a limited modelling power. It has been shown in (Pecina and Schlesinger, 2006) and (Ramisch et al., 2010) that combining several AMs together with other features and using machine learning methods to train a classifier can improve CE. The goal of this paper is to further explore this classification approach for CE in Croatian. Several learning methods are evaluated in an effort to find both the optimal classification model and optimal features using feature subset selection (FSS). In addition to several commonly used traditional features, we also explore the possible benefits of using semantic relatedness between words. Motivated by the future application of our work in terminology and keyword extraction for Croatian, we focus on noun phrases (NP) exclusively. The evaluation is done intrinsically on a set of examples derived from a corpus in Croatian language.

The rest of the paper is structured as follows. In the next section we briefly discuss related work. In Section III we describe the classification methods and features. Section IV presents the experimental setup and evaluation results. Section V concludes the paper and outlines future work.

2. Related Work

Among the first to use lexical AMs based on statistics and information theory were Church and Hanks (1990). A lexical AM measures the lexical association between words in a collocation candidate. The higher the AM value, the more likely it is for the candidate to be a collocation. Some traditional AMs are as follows.

The Dice coefficient is a simple yet remarkably effective measure, which gives larger values for words that often occur together:

$$DICE = \frac{2f(w_1 w_2)}{f(w_1) + f(w_2)} \tag{1}$$

Pointwise mutual information (PMI) is based on information theory and can be viewed as measuring how much in-

formation is shared between the words:

$$PMI = \log_2 \frac{f(w_1 w_2)}{f(w_1) \times f(w_2)} \quad (2)$$

The statistical χ^2 (chi-square) measure is based on testing the hypothesis that the words of a collocation candidate occur independently (Manning and Schütze, 1999):

$$\chi^2 = \sum_{i,j} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \quad (3)$$

Quantities $O_{i,j}$ and $E_{i,j}$ are the actual and expected probabilities of occurrence. These can be obtained using maximum likelihood estimates based on frequency counts.

The above measures were defined for bigrams. In order to improve AM performance on n-grams longer than two words, specialized extension patterns were introduced in (Petrović et al., 2010). For generalization from bigrams to n-grams for (1) and (2) we use the same expressions as (Ramisch et al., 2010). Measure (3) generalizes to n-grams trivially.

A comprehensive evaluation of possible AMs can be found in (Pecina, 2005). There have been several attempts to improve lexical AMs using machine learning. An approach used in (Šnajder et al., 2008) uses genetic programming to evolve optimal AMs for a given training set. Collocation extraction has been treated as a classification problem with AMs as input features in (Pecina and Schlesinger, 2006). Similar features are used in (Ramisch et al., 2010) in addition to basic part-of-speech (POS) information. In contrast to (Pecina and Schlesinger, 2006) and (Ramisch et al., 2010), we explore a new feature type (semantic relatedness between n-gram words). Furthermore, we use wrapper FSS to determine the optimal features for each classifier. The main advantage of such an approach is that it takes into account the way the learning algorithm and the data set interact (Kohavi and John, 1997). This enables us to better understand which features are relevant for identifying collocations.

3. Classification Methods and Features

The classifiers we use include decision trees (C4.5), rule induction (RIPPER), naive Bayes, neural networks, and support vector machines (SVM) with both linear and polynomial kernel. With this list we feel that we have covered a variety of commonly used methods: generative, discriminative, probabilistic, and nonparametric.

We use features already used in similar work (Pecina and Schlesinger, 2006; Ramisch et al., 2010). In addition we introduce some semantically based features. A summary of all features we use is given in Table 1.

3.1. Frequency Counts

The number of occurrences of an n-gram and all subsequences of an n-gram. These are a simple and intuitive choice for a feature since they are obviously important in deciding if a given candidate is a collocation. E.g., for an n-gram $w_1 w_2 w_3$ we use the following counts as features: $f_{w_1}, f_{w_2}, f_{w_3}, f_{w_1 w_2}, f_{w_2 w_3}$, and $f_{w_1 w_2 w_3}$.

Table 1: Summary of used features

Feature class	Description
Frequency counts	Number of occurrences of an n-gram or subsequences of an n-gram
Traditional AMs	Pre-calculated traditional AM values
POS tags	Binary features representing POS information
Semantic	Semantic relatedness of words forming an n-gram

Table 2: Descriptions of POS tags

Tag	Description
N	Noun
A	Adjective
E	Pronouns and numbers
C	Conjunction
S	Preposition
R	Adverbs

3.2. Traditional Lexical AMs

Clearly, lexical AMs provide valuable information for our classifier. In our experiments we use *Dice*, *PMI*, and χ^2.

3.3. Part of Speech

POS of words in n-grams is also used as a feature. For each word w_i in an n-gram there are six binary POS features $P_{i,t}$. Each $P_{i,t}$ is true if and only if the word w_i of the n-gram has POS tag t. The tags used and their meaning is given in Table 2. Note that there is no tag for the remaining word classes in Croatian (Verbs, Interjections, Particles) because NPs of the length we considered almost never contain these word types. To keep the tagset size small, pronouns and numbers were combined into a single class because in the NPs we consider they have virtually identical roles.

3.4. Semantic Features

Semantic features are defined as semantic similarities of all word pairs in an n-gram. E.g., an n-gram $w_1 w_2 w_3$ would have the following features: $s(w_1, w_2), s(w_2, w_3), s(w_1, w_3)$, with $s(w_i, w_j)$ being a semantic similarity measure, which can be modelled in various ways.

We can intuitively justify these features by arguing that semantic relatedness is correlated to the property of being a collocation to a certain degree. Many collocations, such as *"state official"* and *"economy crisis"*, consist of words that have a certain degree of semantic relatedness. Of course we do not expect this to always be the case. In fact, for idioms such as *"hot dog"* the correlation should be negative. Still we hypothesize that machine learning methods

could perhaps benefit from such features. To determine if this hypothesis is true is one of the goals of this paper.

To explore the benefits of using these features in our CE task, a model for semantic similarity is required. For this purpose we employ latent semantic analysis (LSA) (Deerwester et al., 1990). We leave experiments with various other available semantic models for future work. LSA is a well-known mathematical technique based on linear algebra, which can be used to model semantic relatedness. The procedure is summarized as follows.

First we construct a word-document matrix. This is a matrix whose rows correspond to words and columns correspond to documents. The most commonly used method for setting the values of the elements is to set them to the *tf-idf* value of the corresponding word-document pair. Another method, which has been shown to work quite well in (Landauer, 2007), is to use the logarithmic value of word-document frequency and the global word entropy (entropy of word frequency in all documents), as follows:

$$a_{w,d} = \log\left(tf_{w,d}+1\right)\left(1+\frac{1}{\log N}\sum_{d'\in C}\frac{tf_{w,d'}}{gf_w}\log\frac{tf_{w,d'}}{gf_w}\right) \quad (4)$$

where $tf_{w,d}$ value represents occurrence frequency of word w in document d, value gf_w represents the global frequency of word w in corpus C and N is the number of documents in corpus C. Next, singular value decomposition (SVD) is applied to the matrix A yielding two matrices U and V containing left and right singular vectors of A. Finally a dimensionality reduction is performed that approximates the original matrix by keeping only the first k singular values and the corresponding singular vectors (first k columns of U and first k rows of V). This reduction can be interpreted as a removal of noise. Each row of such a reduced matrix U describes a word in the corpus. These vectors form a *concept space* and can be compared (e.g., using cosine similarity) to model the semantic relatedness of words.

Since our corpus was a set of sentences, the documents we use for LSA consist of a single sentence. The method used to construct the word-document matrix was log-entropy (Landauer, 2007) and the number k of dimensions to which we reduce is 250.

While for bigrams we use only one semantic feature – $s(w_1, w_2)$ – for trigrams we use three – $s(w_1, w_2)$, $s(w_1, w_3)$, and $s(w_2, w_2)$ – so it is possible to analyze their correlation using Pearson's coefficient. It is interesting that these pairwise correlations are higher for collocation trigrams (0.365, 0.310, 0.143) than for non-collocation trigrams (0.244, 0.0, -0.004). This is not unexpected, as, on average, words within collocations are more semantically related than words occuring in random n-grams.

4. Evaluation and Results

4.1. Data Set

A corpus was generated by sampling sentences from the Croatian newspaper "Glas Slavonije". The corpus was lemmatized using an automatically acquired morphological lexicon described by Šnajder et al. (2008). A random sample of 1000 bigrams was extracted from the corpus and manually POS tagged. Frequency statistics for each of the bigrams were collected from the lemmatized corpus. Six annotators were given the samples and instructed to annotate those n-grams which they consider to be collocations. The inter-annotator agreement was measured using the κ coefficient with the goal of obtaining an annotated subset with sufficient agreement. Because the main intended application of this work is terminology extraction, we decided to focus on NPs exclusively. Consequently, we manually filtered all non-NPs from the data set. This step could also have been done automatically using the morphological lexicon from (Šnajder et al., 2008).

Table 3: The κ coefficient for bigram collocations

$\kappa(x,y)$	A	B	C	D	E	F
A	–	**0.62**	0.53	**0.61**	0.52	**0.63**
B	**0.62**	–	0.56	**0.73**	0.50	**0.64**
C	0.53	0.56	–	0.55	0.54	0.58
D	**0.61**	**0.73**	0.55	–	0.50	**0.65**
E	0.52	0.50	0.54	0.50	–	0.59
F	**0.63**	**0.64**	0.58	**0.65**	0.59	–

Table 4: The κ coefficient for trigram collocations

$\kappa(x,y)$	A	B	C	D	E	F
A	–	0.38	0.28	0.31	0.32	0.35
B	0.38	–	0.31	0.41	0.40	0.35
C	0.28	0.31	–	0.26	**0.55**	0.32
D	0.31	0.41	0.26	–	0.31	0.47
E	0.32	0.40	**0.55**	0.31	–	0.35
F	0.35	0.35	0.32	0.47	0.35	–

The κ coefficient for bigrams is given in Table 3. Four annotators (A, B, D, and F) had substantial inter-annotator agreement (κ larger than 0.6) and their lists were combined into a bigram data set, resulting in a set of 694 bigrams. Finally, after manually filtering out non-NPs, 534 bigrams remained, 84 (15.7%) of which were labeled as collocations. Values of κ for trigrams are given in Table 4. Even though no pair of samples satisfy the sufficient agreement condition, the experiment was conducted on the pair C and E. This combination yielded a sample of 792 trigrams. After the manual removal of non-NPs, 614 trigrams remained, 239 (38.9%) of which were labeled as collocations.

The observed inter-annotator agreement indicates that the task of determining whether an n-gram is a collocation is quite subjective and the exact boundary is fuzzy even for humans (Krenn and Evert, 2001).

4.2. Evaluation Methodology

It is known that having additional features need not necessarily improve classification performance. Such features can even bring noise into the data and downgrade results. This is why we attempt to find the optimal feature subset.

To this end we use the wrapper FSS approach with the

Table 5: Results for bigram classification

	All features			Feature subset selection		
	Precision	Recall	F1	Precision	Recall	F1
Baseline	70.7 ± 12.6	64.3 ± 15.6	67.3	71.1 ± 7.3	64.2 ± 7.1	67.5
Decision tree	69.2 ± 13.0	67.7 ± 6.8	68.4	75.0 ± 9.1	65.2 ± 5.6	69.8
RIPPER	70.6 ± 7.5	68.8 ± 13.2	69.6	72.3 ± 14.8	61.9 ± 5.2	66.7
Naive Bayes	39.3 ± 8.0	95.2 ± 2.4	55.7	72.5 ± 8.4	77.6 ± 9.3	75.0
Logistic regression	77.6 ± 9.2	78.7 ± 6.9	**78.2**	85.3 ± 13.6	75.0 ± 6.6	**79.8**
Neural network	84.2 ± 10.5	72.7 ± 5.7	78.0	83.4 ± 8.6	72.6 ± 5.9	77.6
SVM (linear)	65.7 ± 9.5	82.2 ± 5.1	73.0	85.5 ± 11.6	70 ± 4.9	76.7
SVM (polynomial)	85.9 ± 6.7	71.3 ± 6.3	78.1	91.5 ± 6.7	67.3 ± 5.1	77.6

Table 6: Results for trigram classification

	All features			Feature subset selection		
	Precision	Recall	F1	Precision	Recall	F1
Baseline	59.2 ± 1.5	62.85 ± 12.4	61.0	49.9 ± 9.4	67.3 ± 14.9	57.3
Decision Tree	61.1 ± 4.2	75.4 ± 6.8	**67.5**	64.9 ± 5.5	68.7 ± 13.4	**66.8**
RIPPER	58.1 ± 3.9	48.3 ± 5.4	52.8	64.9 ± 4.6	56.0 ± 13.4	60.1
Naive Bayes	50.6 ± 1.4	94.4 ± 2.2	65.9	67.9 ± 4.3	61.7 ± 6.5	64.6
Logistic regression	74.8 ± 5.3	52.2 ± 4.7	61.5	68.9 ± 7.3	57.0 ± 11.7	62.4
SVM (linear)	70.7 ± 7.9	58.7 ± 10.9	64.1	70.4 ± 7.2	53.9 ± 8.8	61.1

forward selection algorithm described by Kohavi and John (1997). The algorithm starts with an empty set of features and then it iteratively adds new features. In each iteration the feature that improves performance the most is added to the feature set. The process stops when no remaining feature would provide significant improvement when added to the feature set. This algorithm was chosen because we expect the relevant subset of features to be small with respect to the total number of features.

An important advantage of the wrapper approach to FSS is that it implicitly takes into account redundancy and correlation between features, unlike univariate filter FSS methods. The disadvantage of the wrapper approach is that it is prone to overfitting. In order to prevent overtraining, the entire parameter optimization and FSS procedure is encapsulated in an outer cross validation loop, making it a nested cross validation. The outer validation loop is done in five folds and the inner one in ten folds. E.g., for bigrams the inner loop uses a train set consisting of $\sim$60 collocations and $\sim$384 non-collocations and a validation set containing $\sim$6 collocations and $\sim$42 non-collocations. The optimal feature subset as well as parameters can vary in different folds of the outer validation, however we can still measure the overall importance of a given feature by counting how many times it was chosen during the entire feature selection procedure.

The calculation of SVD required for LSA was performed using the SVDLIBC library.[1] Once all the features were calculated, the evaluation process was implemented as a RapidMiner[2] model.

To measure how well our classifiers work we use the standard F1 measure, which is the harmonic mean of precision and recall first introduced by van Rijsbergen (1979). As a baseline we use a perceptron with a single traditional AM value as input (this ammounts to computing the optimal threshold for the AM). Among the three tested traditional AMs, PMI was chosen as the best performing one.

4.3. Results

After each iteration of the outer validation loop, the optimal set of features for that iteration was recorded. The number of times a feature was chosen during the entire procedure is given in Tables 7 and 8 for bigrams and trigrams, respectively. Only features occurring two or more times are listed. The results for bigram and trigram classification with and without using FSS are given in Tables 5 and 6, respectively. In case of bigrams, the LSA-based semantic feature is chosen often, which implies it is useful. Decision trees seem to be able to take advantage of the χ^2 measure better than the other classifiers. Other methods predominantly use a combination of semantic, PMI, and Dice features. SVMs give better precision, while better recall is achieved by the Bayes classifier. This may indicate that further improvement is possible by using a classifier ensemble. POS tag features are also selected often, especially $P_{1,E}$, which determines if the first word is a pronoun or a number. This is

[1] `http://tedlab.mit.edu/~dr/SVDLIBC/`

[2] `http://www.rapidminer.com`

Table 7: Features used most often for bigram classification

	5x	4x	3x	2x
Baseline	–	–	–	–
Decision Tree	–	χ^2	$s(w_1, w_2)$	$f_{w_1}, f_{w_1 w_2}, P_{1A}, P_{1E}$
RIPPER	–	f_{w_2}, χ^2	$s(w_1, w_2)$	pmi
Naive Bayes	pmi	P_{1A}	P_{2R}	–
Logistic regression	$P_{1N}, s(w_1, w_2)$	P_{1A}	$f_{w_1 w_2}$	P_{1E}
Neural network	$P_{1A}, pmi, s(w_1, w_2)$	$f_{w_1 w_2}$	f_{w_2}, P_{2E}	–
SVM (linear)	$P_{1E}, s(w_1, w_2), pmi$	–	$f_{w_2}, f_{w_1 w_2}$	$dice$
SVM (polynomial)	$P_{1E}, s(w_1, w_2), pmi$	–	$dice, f_{w_2}, f_{w_1 w_2}, P_{2R}$	–

Table 8: Features used most often for trigram classification

	5x	4x	3x	2x
Baseline	–	–	–	–
Decision Tree	–	f_{w_2}	–	
RIPPER	–	f_{w_2}	P_{1E}, P_{2A}, P_{2E}	P_{2N}
Naive Bayes	$P_{2A}, s(w_2, w_3)$	pmi	P_{2R}	$f_{w_3}, P_{2R}, s(w_1, w_3)$
Logistic regression	–	P_{1E}, P_{2A}	P_{1N}, P_{2N}, pmi	$P_{1A}, P_{2E}, P_{3E}, P_{3C}, P_{3R}, P_{2E}, dice$
Bayes net	P_{2A}	$f_{w_2}, s(w_2, w_3)$	–	$f_{w_2 w_3}$
SVM (linear)	P_{1E}, P_{2A}	$dice$	pmi	f_{w_3}, P_{2E}, P_{2C}

along the lines of results obtained by Petrović et al. (2010). In general, in case of bigrams, classifiers using FSS outperform classfiers trained on all features.

Trigram classification appears to be a harder problem and FSS does not seem to be as useful as in the case of bigrams. However, there are some patterns that can be observed. POS features are used by all classifiers. The P_{2A} (second word is an adjective) in particular was selected very often for most of the classifiers. From the selection of other POS features it can be concluded that the *adjective* and *pronoun or number* (which behave very similarly to adjectives) features were selected often. Of classical AMs, *PMI* is the one chosen most often. Classifiers that did not choose classical AMs as features compensated for this by choosing raw frequency features instead. An interesting finding was the performance of the Decision tree classifier, which had a very good result using only the f_{w_2} (frequency of the second word) feature consistently. In addition to f_{w_2}, other features were used in different folds of the outer validation, but each one no more than once. This is not completely unexpected as some of our features are highly correlated.

It is difficult to say which classifier is the best considering the large variances caused mostly by the small size of the data set. Further statistical analysis of the results is required. While there are similar approaches used for English (Pecina and Schlesinger, 2006; Ramisch et al., 2010), to our knowledge, the work reported here is the first attempt to treat collocation extraction in Croatian as a classification problem. Consequently, comparison to existing work in Croatian for collocations (Šnajder et al., 2008; Petrović et al., 2010) is difficult.

5. Conclusion and Future Work

We have evaluated several common machine learning models on the task of collocation extraction for Croatian. The logistic regression classifier gave the best F1 score for bigrams while the decision tree was best for trigrams. Of all the features that were evaluated, it can be concluded that specific POS features, semantic features, and PMI seem to generally contribute the most to best performing classifiers. In our opinion the approach should be further evaluated on a bigger and more consistent data set.

For future work, we also intend to experiment with other types of features such as morphological, syntactic, and other semantic features. A different venue of research can include modifying the methods to perform ranking (regression) instead of classification. Another idea is to perform evaluation on different types of collocations to determine what features work best for what type.

6. Acknowledgments

We thank the anonymous reviewers for their useful comments. This work has been supported by the Ministry of Science, Education and Sports, Republic of Croatia under the Grant 036-1300646-1986.

7. References

K.W. Church and P. Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

A. Jimeno-Yepes, B. Mclnnes, and A. Aronson. 2011. Collocation analysis for UMLS knowledge-based word sense disambiguation. *BMC bioinformatics*, 12.

P. Jin, X. Sun, Y. Wu, and S. Yu. 2010. Word clustering for collocation-based word sense disambiguation. *Computational Linguistics and Intelligent Text Processing*.

R. Kohavi and G.H. John. 1997. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324.

B. Krenn and S. Evert. 2001. Can we do better than frequency? A case study on extracting pp-verb collocations. In *Proceedings of the ACL Workshop on Collocations*, pages 39–46.

T.K. Landauer. 2007. *Handbook of latent semantic analysis*. Lawrence Erlbaum.

Z. Liu, H. Wang, H. Wu, and S. Li. 2010. Improving statistical machine translation with monolingual collocation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.

C. D. Manning and H. Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA.

J. Šnajder, B. Dalbelo Bašić, and M. Tadić. 2008. Automatic acquisition of inflectional lexica for morphological normalisation. *Information Processing & Management*, 44(5).

P. Pecina and P. Schlesinger. 2006. Combining association measures for collocation extraction. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL '06. Association for Computational Linguistics.

P. Pecina. 2005. An extensive empirical study of collocation extraction methods. In *Proceedings of the ACL Student Research Workshop*.

S. Petrović, J. Šnajder, and B. Dalbelo Bašić. 2010. Extending lexical association measures for collocation extraction. *Computer Speech & Language*, 24(2).

C. Ramisch, A. Villavicencio, and C. Boitet. 2010. mwetoolkit: a framework for multiword expression identification. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

C. J. van Rijsbergen. 1979. *Information Retrieval*. Butterworths, London, 2 edition.

J. Šnajder, B. Dalbelo Bašić, S. Petrović, and I. Sikirić. 2008. Evolving new lexical association measures using genetic programming. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. Association for Computational Linguistics.