Predicting corpus example quality via supervised machine learning

Nikola Ljubešić, Mario Peronja

Department of Information and Communication Sciences, Faculty of Humanities and Social Sciences
University of Zagreb, Ivana Lučića 3, HR-10000 Zagreb
{nljubesi,mperonja}@ffzg.hr

Abstract

In this paper we present a supervised-learning approach to extracting good dictionary examples from corpora. We train our predictor of quality on a dataset of corpus examples annotated with a four-level ordinal variable, ranging from a very bad to a very good example. Each of the examples is formally described through 23 variables; the dependence of the quality of which is modelled using a regression model. The evaluation of the ranked results for each of the collocations in the annotated dataset shows that we obtain precision on 10 top-ranked examples of ~80% and a precision of ~90% on the three top-ranked examples. Our approach is highly language independent as well, suffering almost no loss on the 10 top-ranked examples and a loss of ~4% on the three highest-ranked examples once the language-dependent and knowledge-source-dependent features are removed.

Keywords: dictionary example; corpus extraction; supervised machine learning

1. Introduction

Corpus examples are a very welcome part of a dictionary entry. If a dictionary entry includes an example which is a good match for the context in which the user has encountered a word, or for the context in which they want to use it, then the user generally gets what they want in a quick and straightforward way. (Kilgarriff et al., 2008)

Finding good examples manually by looking through concordances in a corpus is very tedious and ranking concordances by the automatically estimated quality of the example is a very welcome addition to lexicographic processes.

The best known tool for finding good examples from a corpus is GDEX (Kilgarriff et al., 2008), part of Sketch Engine (Kilgarriff et al., 2004) where the lexicographer defines criteria for good examples using variables like sentence length, word frequency, pronouns, start and ending of a sentence etc., and has been adapted for a series of languages (Kosem et al., 2011).

In this paper we propose predicting the quality of a corpus example through the paradigm of supervised machine learning where we:

1. manually annotate a sample of examples for a given headword / collocation with it's corresponding quality,
2. define features we consider informative for predicting the quality of a corpus example,
3. train a predictor, using features as explanatory variables and the manually assigned quality as our response variable, and finally
4. use that predictor to rank corpus examples of a headword / collocation by descending predicted quality of the examples.

Beside the prediction task, we measure the informativeness of each feature with the goal of better understanding the underlying phenomenon of what makes a good dictionary example extracted from a corpus.

We run our machine learning experiments by writing feature extractors in Python and performing all supervised learning tasks in scikit-learn (Pedregosa et al., 2011).

2. Dataset

The *conditio sine qua non* of our approach to predicting dictionary example quality is the sample of corpus examples, each of which is human-annotated with a quality score. On this dataset we extract variables, i.e. features we consider informative for predicting the quality of a corpus example for dictionary use. We use those variables and human scores to perform supervised machine learning, i.e. statistical modelling, in which we model the dependence of the response variable (the quality of an example) to the explanatory variables (the features extracted from each of the corpus examples) with the idea of predicting the quality of previously unseen corpus examples.

We extracted our corpus examples from the web corpus of Croatian (Ljubešić and Klubička, 2014). To produce a real-world-scenario sample, we built the dataset from sentences containing one of the 16 collocations chosen as a basis for building this dataset. The collocations were sampled from the hrMWELex lexicon of Croatian multiword expressions (Ljubešić et al., 2014). These 16 collocations consist of four mid-frequency lexemes, each belonging to an open-class part-of-speech: noun, verb, adjective and adverb. Given that we, as will later be described in detail, use shallow features such as sentence length and number of uppercased tokens for predicting the quality of examples, and therefore do not try to model the deep, semantic criteria for a good example, we consider our dataset to be representative for predicting corpus quality of both collocations and single word units.

We finally produced a dataset of 1094 sentences randomly picked from all the sentences of the corpus containing any of the 16 collocations. Each collocation is thereby covered by 14 to 99 examples, which successfully mimics the scenario of extracting collocation examples from a medium-sized corpus.

It is important to note that, since the web corpus is annotated on the morphosyntactic and dependency syntax level, for each of the chosen sentences we had those two annotation layers at our disposal as well.

We annotated each of the 1094 sentences by the following four-level schema:

- 1 – very bad example, the example is useless
- 2 – bad example, most of the example should be rewritten
- 3 – good example, minor changes are necessary
- 4 – very good example, no changes at all are required

The *very bad* score was given to 14% of sentences, the *bad* score to 41.7% of sentences, while the *good* and *very good* scores were given to 33.3% and 11.1% of sentences respectively. This distribution of scores shows that the human annotator considered more than the half of the corpus examples as bad examples. A likely explanation for such a rather high percentage of examples being perceived as bad is that the data, although cleaned, still comes from the web where different types of noise are present on a regular basis.

3. Features

To be able to perform a quality prediction on our potential dictionary examples, i.e. sentences from a corpus, we have to transform each of those sentences into a set of variables. Given that these variables are used for performing the prediction, we refer to them as explanatory variables or features.

We defined altogether 23 features from three different categories: string-based features encoding properties of text on the string level, corpus-based features measuring the coverage of an example by the most frequent words from a corpus and linguistic features that use the linguistic annotation of the candidate example.

The string-based features are the following:

- sent_len: length of the sentence
- avg_len: average token length
- gte10_perc: percentage of tokens longer or equal to 10 characters
- lt3_perc: percentage of tokens shorter than 3 characters
- alphanum_perc: percentage of tokens being alphanumeric
- alphanumpunc_perc: percentage of tokens being alphanumeric or standard punctuations
- startswithucase: whether the sentence starts with an uppercase letter
- endswithpunc: whether the sentence ends with a punctuation

- diac_perc: percentage of tokens containing diacritics
- lcase_perc: percentage of lowercased tokens
- ucase_perc: percentage of uppercased tokens
- tcase_perc: percentage of titlecased tokens
- headpos_perc: relative position of the start of collocation

The corpus-based features were extracted with the help of a token frequency list compiled from the whole hrWaC web corpus. These are the features:

- mf1k_perc: percentage of tokens among the 1k most frequent corpus tokens
- mf10k_perc: percentage of tokens among the 10k most frequent corpus tokens
- mf100k_perc: percentage of tokens among the 100k most frequent corpus tokens

Finally, the linguistic features calculated from the two annotation layers present in the corpus, and thereby in each of our 1094 annotated examples, are thus:

- pron_perc: percentage of pronoun tokens
- pn_perc: percentage of proper noun tokens
- num_perc: percentage of numeral tokens
- sub_num: number of subordinating conjunctions
- co_num: number of coordinating conjunctions
- subco_num: number of conjunctions
- syntcomplex: syntactic complexity as the average length of the dependency arcs

To obtain the first insight into the informativeness of the features for the task at hand, we calculated the p-values for t-tests on each feature given the response variable transformed to a binary *good example* / *bad example* variable. In other words, for each feature we calculated the probability that the difference in the distribution mean of the feature among the good examples and the distribution mean of the feature among the bad examples occurred by chance. The results are given in Table 1.

Among the string-based features we can observe that the sent_len and endswithpunc features are the strongest predictors of the quality of the example. On the other hand, the only statistically insignificant differences are obtained with the gte10_perc and the tcase_perc features.

In corpus-based features, the coverage by the 1,000 most frequent words is shown to be statistically insignificant as well. As the number of the most frequent words increases, the p-value drops off.

Among the linguistic features, the probability of the difference in the means of the percentage of pronouns among good and bad examples is shown to be at very high 40%, indicating that

string-based	p-value
sent_len	**7.0e-18**
avg_len	5.7e-05
gte10_perc	**0.1087**
lt3_perc	9.9e-05
alphanum_perc	4.1e-09
alphanumpunc_perc	5.1e-05
startswithucase	3.5e-04
endswithpunc	**2.7e-20**
diac_perc	0.0064
lcase_perc	0.0063
ucase_perc	0.0045
tcase_perc	**0.0760**
headpos_perc	0.0007

corpus-based	p-value
mf1k_perc	**0.0687**
mf10k_perc	0.0008
mf100k_perc	1.7e-05

linguistic	p-value
pron_perc	**0.4039**
pn_perc	0.0018
num_perc	0.0037
sub_num	5.7e-08
co_num	**7.4e-16**
subco_num	1.3e-15
syntcomplex	8.2e-12

Table 1: T-test p-values for each feature calculated on the feature distributions of good and bad examples

this feature is a bad predictor of the quality of an example. On the other hand, the number of coordinating conjunctions is shown to be a very good predictor. It is interesting to observe that the syntactic complexity of the example has also a very low p-value. One has to be cautious about drawing the conclusion that it is a strong predictor of example quality as it correlates very strongly (0.82) with the feature encoding the sentence length which has an even lower p-value.

4. Experiments and results

The first experiment focused on optimising our regressor. We performed a randomised search hyperparameter optimisation of our Random Forest regressor by doing 10-fold cross-validation. Our scoring function on the regressor was mean absolute error, i.e. the average absolute difference between the human-given quality and the predicted quality. The optimised regressor misses the human score on average by 0.52 points, while the non-optimised regressor produces a mean absolute error of 0.55 points.

In the second set of experiments we measured the ranking performance of our optimised regressor. We evaluated the ranked results via the precision-at-N metric which calculates the precision of the N highest ranked examples. We consider good and very good examples to be positive results and the bad and very bad examples to be negative results.

Since there are examples for 16 different collocates, we ran 16 iterations, during each we trained our regressor on examples of 15 collocates, and used the regressor to produce the ranked result for the left-out collocate examples. We calculated the final precision as the arithmetic means of the precisions of each collocate.

We compared the obtained result with a baseline system which orders the examples randomly and a ceiling system which orders the examples by the score given by the human annotator.

The results of this set of experiments are presented in Table 2. While the baseline gives a precision of around 50%, as expected, given the distribution of scores in the annotated dataset, the ceiling shows that each of the 16 collocations has at least five good examples, while the precision drops slightly when we consider the 10 highest-ranked examples of each collocate.

The result obtained with the four-level regressor is regressor_4. It has precision of ∼80% to ∼90%, depending on the number of candidates taken into account, which is much closer to the ceiling than to the baseline.

The regressor_2 system is the one trained on two levels of the response variable only, i.e. it does not use the information about the difference between good and very good examples on one side, and bad and very bad examples on the other side. We can observe a minimal drop, showing that manually annotating the data with a two-level categorical variable is almost as informative for this task as our four-level ordinal variable.

	P@10	P@5	P@3
baseline	48.7%	48.7%	48.7%
ceiling	98.8%	100.0%	100.0%
regressor_4	78.8%	86.6%	89.3%
regressor_2	78.2%	86.2%	89.1%

Table 2: Precision on first N candidates obtained with the random baseline, the ceiling, and a regressor trained on 4-level and 2-level response variables

In the next experiment we considered using subsets of features only. We envisaged the following scenarios:

- regressor – using all features
- regressor_string – using string features only, i.e. not having (large) corpora at our disposal and the possibility of a linguistic analysis
- regressor_langind – using string features only without the percentage of diacritics as it could be considered specific for the Croatian language, thereby assessing how well our system could work on any other language

The results are presented in Table 3. The drop is surprisingly low when removing outer knowledge sources like the corpus and tools for linguistic analysis, showing a minor drop if 10 candidates are taken into account and a 3.7% drop on the first three candidates. Making the predictor language-independent adds an additional below-1% loss. It is important to

stress that the language-independent predictor would still need annotated data in the other language. Measuring the predictor performance in another language without retraining it on the data of that language could be very interesting and is left, as we do not have testing data for other languages, for future work.

	P@10	P@5	P@3
regressor	78.8%	86.6%	89.3%
regressor_string	79.0%	83.9%	85.7%
regressor_langind	78.4%	83.2%	85.0%

Table 3: Precision on first N candidates obtained with the regressor using all features, the regressor using string features only and the language-independent regressor

We finally depict the probability distribution of the examples of a specific quality obtained when using the baseline, and when taking into account the first 10, five or three top-ranked examples. These distributions are given in Figure 1.

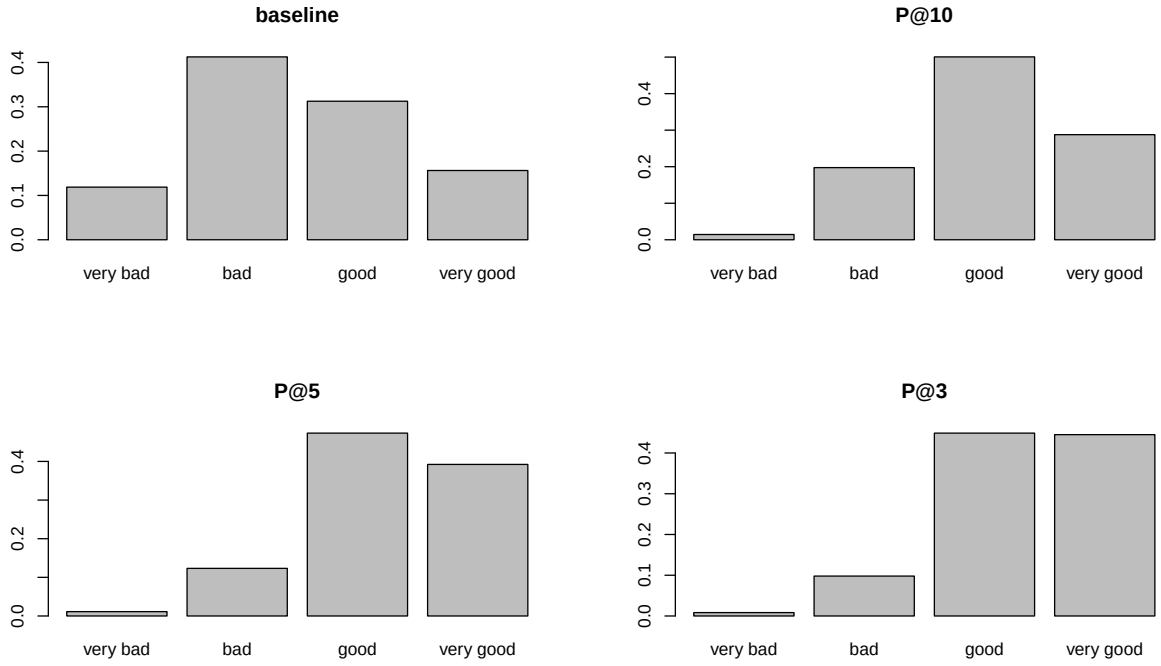

Figure 1: Probability distribution of the quality of the examples for the random baseline and the system taking into account first 10, 5 and 3 top-ranked examples

Having a better ratio between good and very good examples as we consider a lower number of highest-ranked candidates is expected. It points to the conclusion that the ranker manages

to produce the best results on the top of each output and that the results deteriorate as we move down the ranked output.

We can observe that we drastically outperform our baseline. While the best represented category in the baseline are bad examples, in P@10 and P@5 the good category is the most prominent one, the P@3 output having a similar amount of good and very good examples in the output.

5. Conclusion

In this paper we have presented an approach to extracting good corpus examples for dictionary use by using supervised learning, i.e. building a prediction model on a dataset on which the corpus example quality was already attested by a human. We argue that this approach is much more convenient than that used in GDEX where a lexicographer defines criteria for good examples by hand. Specifically, examples have to be annotated, or chosen, anyway, and such prediction algorithms have a steep learning curve, meaning that after annotating just a few examples, the ranking of the candidate examples improves drastically.

We have inspected the informativeness of each of the features used, showing that shallow features, such as the length of the example and the use of punctuation, and some less shallow features that are dependent on the shallow ones, such as the number of coordinating conjunctions, is most informative for the task.

In the ranking experiments we have shown to produce precision of ~80% on the first 10 candidates and ~90% on the first three candidates, which outperforms the random baseline of ~50% precision drastically.

We have shown that removing all external information sources, such as the corpus and its annotation, and language-dependent features, such as the percentage of diacritics, deteriorates our results among the first three top-ranked candidates slightly, lowering precision by ~4%.

Our future work will involve two main directions of research. The first direction is testing the system on different languages and checking the language independence of the approach in both cases, when training data (i.e. annotated examples) in the new language is present, or when it is not and the model built on one language is applied directly onto the sentences of another language.

The second direction of our future work is the comparison of this approach with the rule-based approach, such as GDEX, where the (probably computational) lexicographer defines the criteria for a good dictionary example by hand.

6. Acknowledgements

The research leading to these results has received funding from the European Fund for Regional Development 2007-2013 under grant agreement no. RC.2.2.08-0050 (project RAPUT).

7. References

Kilgarriff, A., Husák, M., Rundell, M., McAdam, K. & Rychlý, P. (2008). GDEX: Automatically finding good dictionary examples in a corpus. In *Proceedings of the XIII EURALEX International Congress*, Barcelona. Institut Universitari de Lingüística Aplicada. pp. 425–432.

Kilgarriff, A., Rychlý, P., Smrž, P. & Tugwell, D. (2004). The Sketch Engine. *Information Technology*, 105:116.

Kosem, I., Husák, M. & McCarthy, D. (2011). GDEX for Slovene. In *Electronic lexicography in the 21st century: New Applications for New Users: Proceedings of eLex 2011, Bled, 10-12 November 2011*. pp. 151–159.

Ljubešić, N., Dobrovoljc, K., Krek, S., Antonić, M. P. & Fišer, D. (2014). hrMWELex – A MWE lexicon of Croatian extracted from a parsed gigacorpus. In Erjavec, T. and Gros, J. Ž., editors, *Language technologies: Proceedings of the 17th International Multiconference Information Society IS2014*, Ljubljana, Slovenia.

Ljubešić, N. & Klubička, F. (2014). {bs,hr,sr}WaC – web corpora of Bosnian, Croatian and Serbian. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, Gothenburg, Sweden. Association for Computational Linguistics. pp. 29–35.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, pp. 2825–2830.