

The stability of the parameter transformation with Zipfian distributions across corpora

Balázs Indig

Pázmány Péter Catholic University, Faculty of Information Technology and Bionics
MTA-PPKE Hungarian Language Technology Research Group
50/a Práter Street, 1083 Budapest, Hungary
`indig.balazs@itk.ppke.hu`

Abstract. Nowadays most of the NLP methods require the tuning of some parameters. This fact implies that training data must be split into a development set and a training set to optimise the parameters carefully before testing. The problem arises from the differences between the development and the test set. Indig [5] suggests that for English language NP-chunking – and probably other tasks as well – one can use the Zipfian distribution of the development set and test set to transform the parameters from the former to the latter. The mayor weakness of his theory is that it is supported by only one measurement and lacks the comparison of other transforming methods. In this paper we test rigorously if Indig’s statement on the parameter transformation is correct for different corpus sizes as well. We also compare the Zipfian distribution to similar but simpler features. We use the *CoNLL-2000* corpus for arbitrary phrase chunking to get comparable results.

Keywords: parameter optimisation, phrase chunking, sequential tagging

1 Introduction

Arbitrary phrase chunking, also known as *shallow parsing* is a well-known *bracketing problem* which is most likely to be solved with *sequential tagging*. For English the *CoNLL-2000 dataset* [11] is the de facto standard dataset to measure tagger performance, that is why we have chosen it for the corpus of our measurements. The current state-of-the-art method, *Less is more* [5] has improved the previous state-of-the-art *Gut, Besser, Chunker* [6] by setting the threshold from 50 to 13 and determined the optimal number by parameter transition between the development set and the test set. The underlying method uses the same *mild lexicalisation* which was introduced in *Gut, Besser, Chunker*. There are multiple methods described in *Less is more*, however, the main difference from its predecessors is a ‘test-set optimised’ parameter resulting in a specialised model created from the ‘general optimised’ parameters which were obtained on the development set. This method achieves the F-score of 95.53% (96.69% for NPs) with the IOBES representation, which the authors consider as a result close to the

theoretical maxima, which implies that in this paper we do not plan to overcome these results, but thoroughly test the stability of the method instead.

Both of the aforementioned methods are based on *CRFsuite* [9], a simple linear-chain conditional random field (CRF) tagger¹. We must note that the use of this simple tagger with smart transformation of the input and output before and after tagging is proven better than a sophisticated tagger with a *bidirectional LSTM-CRF model* [4] alone which only achieves 94.46 % F-score. Another interesting fact – shown empirically by Indig and Endrédi [6] – is that there is no extra information in using and voting between different representations of IOB sequences as suggested by Shen and Sarkar [10]. Therefore in our measurements we only check the IOBES representation as it is empirically verified that it has superior performance to the other four.

Indig [5] used the uniform shape of the underlying Zipfian distributions of the development and test sets to get better parameter values automatically. He suggested that the method could be utilised for other tasks like POS tagging and NER as well, as word frequency is naturally available in all tasks and languages, but he evaluated the method on English phrase chunking only with one splitting for different IOB representations. The mayor flaw in the design of the experiment was – as stated previously in the paper – that IOB representations are equal from the examined perspective.

The paper is organised as follows: after a brief introduction to the topic we present multiple measurements on the same corpora with different sizes and splits and compare the parameter estimation power of the difference of the underlying Zipfian distributions to other simpler features like the size difference of the corpus in tokens and types.

2 Chunking nowadays

Chunking in fact is a *labelled bracketing problem* where each label (B=[, E=], I=inside, O=outside, S=single) – which denotes the actual state of the brackets – has an additional identifier to mark the class of words that the content of the bracket belongs to. For NP-chunking it corresponds to the subtree of the parse tree of the sentence that contains the bracketed phrase. The main advantage of this method to real parsing is that chunks can be assigned to tokens with less computational effort compared to a full-fledged accurate parser. Moreover, many languages still lack this kind of parser, but have rather small or medium-sized, manually annotated corpora which can be used for sequential tagging. The other advantage of the method is that one can solve tasks including but not limited to Part-of-Speech (POS) tagging and Named-entity recognition (NER) with the same tagger framework. For most of the languages, the aforementioned problems have an off-the-shelf solution which can be used to overcome the lack of a real parser.

¹ This tagger achieves 93.79% without lexicalisation for arbitrary phrase chunking on the IOBES representation

Traditionally, the number of classes had to be reduced to speed up the process, but this minor improvement can be safely ignored nowadays. Reducing the number of labels introduces two problems: (a) the decrease of accuracy because it became harder for the tagger to decide between the remaining rather complex classes, (b) Indig and Endr edy [6] as well as Indig [5] point out that with the reduced number of classes (which comes from different IOB representation) and also the extended number of classes (which comes from lexicalisation) damage the structure of the tag sequences – in other words the well-formedness –, which was ignored to that point in the literature.

Indig and Endr edy states that the conversion between different IOB representations is not trivial and is the most flawed part of the previous state-of-the-art method [10]. They also state that if one has to convert between representations, he or she should use the Stanford CoreNLP’s IOBUtils [7] which has the proper converter available to reliably convert between each representation and as a side effect also fix well-formedness issues. Our target method, *Less is more* also used this feature in the measurements so we will follow this path.

2.1 Lexicalisation

Molina and Pla [8] invented a lexicalisation method (in this paper we call it *full lexicalisation*) which was thoroughly investigated by Indig and Endr edy [6]. They invented a lighter variant with less labels (*mild lexicalisation*) which had superior performance in their experiments resulting in a new-state-of-the-art method (see Table 1. for the comparison of the different lexicalisation variants). Later in *Less is more* this variant was used with different thresholds.

Table 1. *Mild lexicalisation*: only IOB labels of the words above a given frequency threshold are augmented with the word and with its POS tag, otherwise the fields are left untouched. (‘+’ sign is used as a separator)

	Unlexicalised		Lexicalised			
	<i>Original format</i>		<i>Full</i>		<i>Mild (just words)</i>	
Word	POS	IOB Label	POS	IOB Label	POS	IOB Label
Rockwell	NNP	B-NP	NNP	NNP+B-NP	NNP	B-NP
said	VBD	O	VBD	O	VBD	O
the	DT	B-NP	the+DT	the+DT+B-NP	the+DT	the+DT+B-NP
agreement	NN	I-NP	NN	NN+I-NP	NN	I-NP

The authors’ original intention with *Gut, Besser, Chunker* was to use the lighter lexicalisation form to adapt the method for agglutinative languages like Hungarian, but they have found that their method – with the reduced tagset – is still not feasible due to the high number of tags. Further research on the optimal threshold [5] had finally lead to a dead end, because the new state-of-the-art results relied on a significantly lower threshold 13 (instead of the previous 50)

largely increasing the number of tags in the tagset. Indig claimed [5] that if one would lexicalise all vocabulary words, the whole problem could be reduced into a POS-tagging problem, with some extra tags added to represent the bracketing, so a totally different approach would be needed to maintain feasible training time.

To visualize the aforementioned problem of the different thresholds in different languages on different sized corpora, one can easily draw something similar to Figure 1.

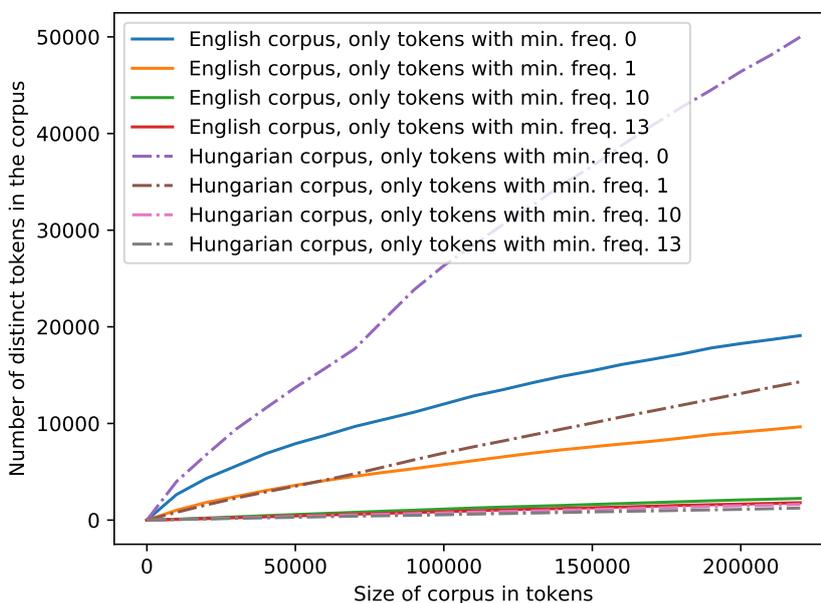


Fig. 1. The new distinct token rate as the function of corpus size for English (CoNLL-2000 training data) and Hungarian (HGC [2]) corpora. Different minimal word frequency thresholds are also marked.

For agglutinative languages – in this case for Hungarian² – the rate of the words are rising even when the English word rate stabilises at larger corpus sizes. The interesting fact is that if we cut down low frequency tokens as done with lexicalisation, one can see that both languages’ word rate stabilises even for small corpora. Interestingly for Hungarian the 13 as threshold yields less distinct tokens compared to the English counterpart. The verification that the

² For the Hungarian corpus we used the first CoNLL-2000 training data sized part of the Hungarian Gigaword Corpus [2].

two corpora yield two Zipfian distribution with different parameters is left for the reader.

The above fact enables us to state that for English the lexicalisation method is stable and feasible with a specific amount of training material, but for agglutinative languages the continuous growth of the vocabulary and the high number of rare tokens do not allow the direct usage of this method – because insane amount of manually created training data would be needed. As we can see that the same threshold value can mean very different values for different sized corpora. This forces one to accommodate the actual threshold for a specific corpus and task. In the literature one can find many forms of using a threshold because it ‘just worked’ for a specific task (for example in POS tagging [1] and in NP-chunking [10] as well). Papers citing the specific paper blindly use the same threshold without a doubt. We argue that in any case one must disclose why that specific threshold was chosen and whether the measurements support it. One threshold does not fit all, therefore sometimes it needs to be transformed as we show it in the next section.

3 Parameter Transformation to Adapt the Model

In machine translation, different domains of text are needed to be translated. In some domain the quantity of the text is limited, so it is impossible to train a full translation system on it. Therefore there is a method of unsupervised adaptation of the general model to the source domain to improve the quality of the translation [3]. One must feed the source side of the test set to the translation system in order to adapt it to the specific test corpus.

This method is very similar to the threshold translation done *Less is more* for arbitrary phrase chunking where the problem with parameter optimisation arises from the difference between the test and development sets. The development set has similar properties as the training set that it is stripped from, but in real life the test set is from another corpus with different frequency distribution of word forms. We used the *CoNLL-2000* dataset and stripped every 10th sentence from the training set for development set resulting in the following three corpora with their respective sizes: the training set is 198,870, the development set is 21,793 and the test set is 49,389 token long. So if one would set an optimal threshold X from the development set, which is bound to the small size of the development set it would not fit to the distribution of the words in the much larger test set.

To address this problem, Indig [5] uses the two underlying Zipfian distributions of the development and the test sets – which have different characteristics – and uses linear optimisation to transform the threshold to the other set using the following formula: $\min_{x,y}(ax + y)$ as it can be seen in Figure 2. The tangent of the optimal threshold level on the Zipfian curve (denoted with a in the formula) is computed using the two neighbouring points of the curve. Indig’s theory is that this tangent should be the same on both curves at the optimal threshold because of the invariant properties of the two Zipfian curves. If one

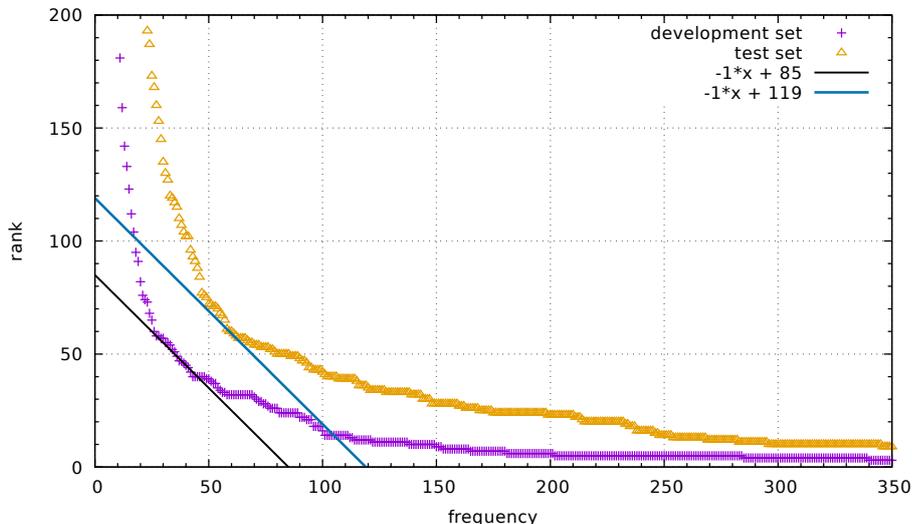


Fig. 2. Zipfian distribution of the development and test sets. The tangent of the curve corresponding to the lexicalisation level producing the maximum score on the development set is transformed to the test set by linear optimization.

would blindly accept X as a threshold it would point to a different tangent of the Zipfian curve which has different properties.

So the transformation is reasonable, but nothing has been said so far about the stability of the method on different corpus sizes or other promising features. In the following sections we compare this heuristic to other simpler methods for obtaining the optimal threshold on the test set. We also test this method with different corpus sizes and splitting.

4 Method

We separated a development set from the CoNLL-2000 training set (in IOBES representation) by using the 10% of sentences as it was created in the aforementioned papers. The difference is that we tested all ten splittings (1st, 2nd, 3rd, etc. sentence was used as development set) and also we reduced the training corpus size to the half to see how it affects the method. We used mild lexicalisation and the CRFsuite tagger. For each of the splits we applied training and test runs for the development sets with different lexicalisation levels around 13 to see which threshold yields the optimal results. The resulting word sets were used for lexicalisation. We lexicalised the three sets according to each lexicalisation level and tagged both the development and the test set yielding two tagging results per lexicalisation level per split. The resulting tagged sets were then delexicalised and the well-formedness of tag sequences was fixed with IOBTools converter of

CoreNLP [7] before evaluation. The gold standard annotation is only revealed in the evaluation step until that point it was treated as non-existent as it is for real life data.

In order to determine the effect of the transformation method, we explored the resulting scores and searched for the best F-score achieved in the *development phase* to set the according lexicalisation level for the measurements afterwards in the *test phase*. This method has been repeated for the different splits and also on the halved training corpus. The last free parameter of the measurement was the transformation algorithm. We tested the following transformation methods:

- Use the same value (raw threshold yield from the development set)
- Use the Zipfian distribution presented in Section 3
- Multiply with the ratio of the distinct token number of the development and the test set
- Multiply with the ratio of the size of the development and the test set

5 Results

For each lexicalisation level we tagged the development set and the test set. We selected the lexicalisation level with the best result for each eleven pair. This method yield two type of result: (a) The best threshold for all split on the development set to work with in the later steps. (b) The ideal solution for all split on the test set which helps to set an upper bound for the examined methods (see Table 2.). In the next step we transformed the threshold obtained from the development set to the test set with all four methods. The transformed lexicalisation levels were checked among the results of the test set and yielded the actual results reachable with the four methods on the splits (see Table 3).

Table 2. The best lexicalisation levels for the development and the test set. The lexicalisation level of the former is the baseline of the transformations and the score of the latter is the ceiling of the method.

	1	2	3	4	5	6	7	8	9	10 (orig)	Halved corpus
development set	94.72 12	93.84 14	94.98 16	95.01 15	94.41 16	94.46 11	94.54 14	95.32 13	95.1 15	94.99 16	95.03 7
test set	94.79 13	94.11 18	95.21 15	94.68 12	94.94 13	95.39 13	94.7 12	95.63 14	95.19 17	95.62 14	95.61 8

As we do not reveal the gold standard annotation of the test set until the final evaluation part, all presented methods (Section 4 and 3) can be used in real life for any test set with or without gold standard data. The optimal parameters (threshold) are set previously on the development set therefore, we do not train

on (the gold standard annotation of) test set. Independently, by transforming ‘blindly’ the optimal threshold gained from the development set to the test using the four methods, we do not utilise (the gold standard annotation of) the test set, just the input corpora – which are available naturally – for the parameter adjustment.

Table 3. The final scores for the test set (with the transformed lexicalisation levels).

	1	2	3	4	5	6	7	8	9	10 (orig)	Halved corpus
raw	94.45	93.39	94.4	93.61	94.13	94.29	94.4	95.24	94.92	94.87	94.82
threshold	12	14	16	15	16	11	14	13	15	16	7
Zipfian distribution	94.76	94.0	95.13	94.63	94.87	95.32	94.62	95.51	95.13	95.53	95.49
distinct token ratio	93.16	93.4	94.25	93.17	93.01	94.57	94.4	94.6	94.29	95.02	92.21
	17	12	17	18	12	14	14	11	16	15	14
corpus size ratio	94.24	93.66	94.4	93.15	93.78	95.32	94.06	94.6	94.92	94.89	93.44
	16	13	16	16	17	15	13	11	15	11	12

In Table 3 – which contains the quantitative analysis of the different lexicalisation models along with lexicalisation thresholds for different splits – one can see that most of the lexicalisation levels that belong to the best F-scores (%) are near 13 which was set by Indig [5]. We can conclude that each lexicalisation level behaves similarly regardless of the tested set, but the behaviour of the tested parameter transition models are quite different. The Zipfian distribution as the transformation method turned out to be the best heuristic and stable even when the training corpus were halved, while the tested simpler methods could not reliably estimate the parameter for the test set. We think that this is because the amount of the encoded information regarding the corpus is lower than in the Zipfian distribution. In the 6th split, the *corpus size ratio* method reached the same score as the Zipfian, but this means it is not stable enough in different conditions. We can conclude that the parameter transformation in *Less is more* is the best of the tested variants yielding the state-of-the-art score (see Table 4) and it is stable across multiple splitting.

Table 4. Summary of final F-scores

chunking method	arbitrary phrases	NPs
Shen and Sarkar [10]	94.01	95.23
Indig and Endrédy [6]	95.06	96.49
Indig [5] (13 as threshold)	95.53	96.69

6 Conclusion and Future Work

We presented a thorough analysis of the transformation of the optimal threshold for multiple splitting with multiple heuristics on English arbitrary phrase chunking as a test case. One of the tested heuristics was the current state-of-the-art method. The examined state-of-the-art method has proven to be better than the other simpler methods compared, as the Zipfian distribution encodes the most important features of the corpus – which were tested – as the number of tokens (as the frequency) and the number of types (as the rank). We can not create a corpus that does not correspond to the Zipfian distribution which implies that changing the frequency of a selected element is bound to the frequency of other elements including the ones that the corpus does not contain.

The best tested method can be used to transform the parameters gained on the development set to the test set taking advantage of the invariant properties of the similarity of the two frequency distributions – which are naturally available – in order to have better and possibly more stable results. Using this transformation, the results of the tagging could be improved near to the global optima of the method. We think lexicalisation level transforming is relevant for other tasks that benefit from finding frequency cut-offs across corpora as well.

Due to the fact that frequency distributions do not behave like normal functions because of the individual differences in the data, the linear optimisation method did not achieve the best F-scores. A higher level approximation that takes this fact into account better and uses more global information could make this method applicable. To adapt the phrase chunking method to agglutinative languages or to develop it further, one must do something more similar to POS-tagging than the traditional approach which may be done in parallel in the future.

References

1. Brants, T.: TnT: a statistical part-of-speech tagger. In: Proceedings of the sixth conference on Applied natural language processing. pp. 224–231. Association for Computational Linguistics (2000)
2. Csaba, O., Tams, V., Blint, S.: The hungarian gigaword corpus. In: Chair), N.C.C., et al. (eds.) Proceedings of the 9th International Conference on Language Resources and Evaluation. ELRA, Reykjavik, Iceland (may 2014)
3. Faražian, M.A., Turchi, M., Negri, M., Federico, M.: Multi-domain neural machine translation through unsupervised adaptation. In: Proceedings of the Second Conference on Machine Translation. pp. 127–137 (2017)
4. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
5. Indig, B.: Less is more, more or less... – finding the optimal threshold for lexicalisation in chunking. In: Computational Linguistics and Intelligent Text Processing: 18th International Conference, CiCLing. pp. (Accepted, in press) (2017), (Accepted, in press)
6. Indig, B., Endrédy, I.: Gut, besser, chunker – selecting the best models for text chunking with voting. In: Computational Linguistics and Intelligent Text Processing: 17th International Conference, CiCLing. pp. (Accepted, in press) (2016)

7. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 55–60 (2014), <http://www.aclweb.org/anthology/P/P14/P14-5010>
8. Molina, A., Pla, F.: Shallow parsing using specialized HMMs. *The Journal of Machine Learning Research* 2, 595–613 (2002)
9. Okazaki, N.: CRFsuite: a fast implementation of Conditional Random Fields (CRFs) (2007), <http://www.chokkan.org/software/crfsuite/>
10. Shen, H., Sarkar, A.: Voting between multiple data representations for text chunking. In: Kégl, B., Lapalme, G. (eds.) *Advances in Artificial Intelligence, 18th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2005, Victoria, Canada, May 9-11, 2005, Proceedings. Lecture Notes in Computer Science*, vol. 3501, pp. 389–400. Springer (2005), [\url{http://dx.doi.org/10.1007/11424918_40}](http://dx.doi.org/10.1007/11424918_40)
11. Tjong, E.F., Sang, K., Buchholz, S.: Introduction to the CoNLL-2000 shared task: Chunking. In: Proceedings of the 2Nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning - Volume 7. pp. 127–132. *ConLL '00*, Association for Computational Linguistics, Stroudsburg, PA, USA (2000), [\url{http://dx.doi.org/10.3115/1117601.1117631}](http://dx.doi.org/10.3115/1117601.1117631)