

Singleton Detection using Word Embeddings and Neural Networks

Hessel Haagsma

University of Groningen

The Netherlands

hessel.haagsma@rug.nl

Abstract

Singleton (or non-coreferential) mentions are a problem for coreference resolution systems, and identifying singletons before mentions are linked improves resolution performance. Here, a singleton detection system based on word embeddings and neural networks is presented, which achieves state-of-the-art performance (79.6% accuracy) on the CoNLL-2012 shared task development set. Extrinsic evaluation with the Stanford and Berkeley coreference resolution systems shows significant improvement for the first, but not for the latter. The results show the potential of using neural networks and word embeddings for improving both singleton detection and coreference resolution.

1 Background

Coreference resolution is the task of identifying and linking all expressions in language which refer to the same entity. It is an essential part of both human language understanding and natural language processing. In NLP, coreference resolution is often approached as a two-part problem: finding all referential expressions (a.k.a. ‘mentions’) in a text, and clustering those mentions that refer to the same entity.

So, in Example (1), the first part consists of finding *My internet*, *It*, and *it*. The second part then consists of clustering *My internet* and *it* together (as indicated by the indices), and not clustering *It* with anything (as indicated by the x).

- (1) [My internet]₁ wasn't working properly.
[It]_x seems that [it]₁ is fixed now, however.

This example also serves to showcase the difficulty of the clustering step, since it is challenging to decide between clustering *My internet* with *it*, clustering *My internet* with *It*, or clustering all three mentions together. However, note that in this sentence *It* is non-referential, i.e. it does not refer to any real world entity. This means that this mention could already be filtered out after the first step, making the clustering a lot easier.

In this paper, we improve mention filtering for coreference resolution by building a system based on word embeddings and neural networks, and evaluate performance both as a stand-alone task and extrinsically with coreference resolution systems.

1.1 Previous Work

Mention filtering is not a new task, and there exists a large body of previous work, ranging from the rule-based non-referential *it* filtering of Paice and Husk (1987) to the machine learning approach to singleton detection by de Marneffe et al. (2015).

Different mention filtering tasks have been tried: filtering out non-referential *it* (Boyd et al., 2005; Bergsma and Yarowsky, 2011), non-anaphoric NPs (Uryupina, 2003), non-antecedent NPs (Uryupina, 2009), discourse-new mentions (Ng and Cardie, 2002), and singletons, i.e. non-coreferential mentions (de Marneffe et al., 2015). All these tasks can be done quite accurately, but since they are only useful as part of an end-to-end coreference resolution system, it is more interesting to look at what is most effective for improving coreference resolution performance.

There is much to gain with improved mention filtering. For example, the authors of one state-of-the-art coreference resolution system estimate that non-referential mentions are the direct cause of 14.8% of their system's error (Lee et al., 2013). The importance of mention detection and filtering

is further exemplified by the fact that several recent systems focus on integrating the processing of mentions and the clustering of mentions into a single model or system (Ma et al., 2014; Peng et al., 2015; Wiseman et al., 2015).

Other lessons regarding mention filtering and coreference resolution performance come from Ng and Cardie (2002) and Byron and Gegg-Harrison (2004). They find that the mentions filtered out by their systems are also the mentions which are least problematic in the clustering phase. As a result, the gain in clustering precision is smaller than expected, and does not compensate for the recall loss. They also find that high precision in mention filtering is more important than high recall.

The state-of-the-art in mention filtering is the system described by de Marneffe et al. (2015), who work on singleton detection. De Marneffe et al. used a logistic regression classifier with both discourse-theoretically inspired semantic features and more superficial features (animacy, NE-type, POS, etc.) to perform singleton detection. They achieve 56% recall and 90% precision on the CoNLL-2012 shared task data, which translates to a coreference resolution performance increase of 0.5-2.0 percentage point in CoNLL F1-score.

1.2 The Current Approach

In this paper, a novel singleton detection system which makes use of word embeddings and neural networks is presented. There are three main motivations for choosing this approach, partly based on lessons drawn from previous work.

The first is that the coreference resolution systems we evaluate with here do not make use of embeddings. Thus, using embeddings as an additional data source can aid in filtering out those singletons which are problematic for the clustering system. Word embeddings are chosen because we expect that the syntactic and semantic information contained in them should help the singleton detection system to generalize over the training data better. For example, knowing that ‘snowing’ is similar to ‘raining’ makes it easier to classify ‘It’ in ‘It is snowing’ as singleton, when only ‘It is raining’ occurs in the training data.

Second, previous work indicated that precision in filtering is more important than recall. Therefore, a singleton detection system should not only be able to filter out singletons with high accuracy, but should also be able to vary the precision/recall

trade-off. Here, the output is a class probability, which fulfils this requirement.

Third, both Bergsma and Yarowsky (2011) and de Marneffe et al. (2015) find that the context words around the mention are an important feature for mention filtering. Context tokens can easily be included in the current set-up, and by using word embeddings generalization on these context words should be improved.

2 Methods

The singleton detection system presented here consists of two main parts: a recursive autoencoder and a multilayer perceptron. The recursive autoencoder is used to create fixed-length representations for multi-word mentions, based on word embeddings. The multi-layer perceptron is used to perform the actual singleton detection.

2.1 Data

We use the OntoNotes corpus (Weischedel et al., 2011; Weischedel et al., 2013), since it was also used in the CoNLL-2011 and 2012 shared tasks on coreference resolution (Pradhan et al., 2011; Pradhan et al., 2012), and is used by de Marneffe et al. (2015). A downside of the OntoNotes corpus is that singletons are not annotated. As such, an extra mention selection step is necessary to recover the singleton mentions from the data.

We solve this problem by simply taking the mentions as they are selected by the Stanford coreference resolution system (Lee et al., 2013), and use this as the full set of mentions. These are similar to the Berkeley coreference resolution system’s mentions (Durrett and Klein, 2013), since they mention they base their mention detection rules on those of Lee et al. This makes them suitable here. In addition, de Marneffe et al. (2015) use the Stanford system’s mentions as basis for their singleton detection experiments, so using these mentions aids comparability as well.

2.2 Recursive Autoencoder

A recursive autoencoder (RAE) is applied to the vector representations of mentions, reducing them to a single word-embedding-length sized vector. This is done to compress the variable length mentions to a fixed-size representation, which is required by the multi-layer perceptron.

The RAE used here is similar to the one used by Socher et al. (2011), with the following de-

sign choices: a sigmoid activation function is used, training is done using stochastic gradient descent, and the weights are untied. A left-branching binary tree structure is used, since only the final mention representation is of interest. Euclidean distance is used as an error measure, with each vector’s error weighted by the number of words it represents.

2.3 Multi-layer Perceptron

The multi-layer perceptron consists of an input layer, one hidden layer, and a binary classification layer. As input, three types of features are used: the mention itself, context words around the mention, and other mentions in the context.

The implementation of the MLP is straightforward. The input order is randomized, to prevent spurious order effects. Stochastic gradient descent is used for training. Experiments with various settings for the parameters governing learning rate, number of training epochs, stopping criteria, hidden layer size, context size and weight regularization are conducted, and their values and optimization are discussed in Section 3.1.

2.4 Integrating Singleton Detection into Coreference Resolution Systems

Two coreference resolution systems are used for the evaluation of singleton detection performance: the Berkeley system (Durrett and Klein, 2013) and the Stanford system (Lee et al., 2013).

The Stanford system is a deterministic rule-based system, in which different rules are applied sequentially. It was the highest scoring coreference resolution system of the CoNLL-2011 shared task. The most natural way of integrating a singleton detection model in this system is by filtering out mentions directly after the mention detection phase.

The Berkeley system, on the other hand, is a learning-based model, which relies on template-based surface-level features. It is currently one of the best-performing coreference resolution systems for English. Because the system is a retrainable learner, the most obvious way to use singleton detection probabilities is as a feature, rather than a filter. For both systems, varying ways of integrating the singleton detection information are presented in Section 3.3.

3 Evaluation & Results

3.1 Preprocessing and Optimization

The recursive autoencoder was trained on the CoNLL-2011 and 2012 training sets, with a learning rate of 0.005. Training was stopped when the lowest validation error was not achieved in the last 25% of epochs. The trained model was then used to generate representations for all mentions in the development and test sets.

Using these mention representations, the parameters of the MLP were optimized on the CoNLL-2012 development set. The stopping criterion was the same as for the RAE, and the learning rate was fixed at 0.001, in order to isolate the influence of other parameters. During optimization, the following default parameter values were used: 50-dimensional embeddings, 150 hidden nodes, 5 context words (both sides), 2 context mentions (both sides), and a 0.5 threshold for classifying a mention as singleton. A competitive baseline was established by tagging each pronoun as coreferential and all other mentions as singleton. We test for significant improvement over the default values using pair-wise approximate randomization tests (Yeh, 2000).

For the hidden layer size, no value from the set $\{50, 100, 300, 600, 800\}$ was significantly better than the default of 150. In order to keep the input/hidden layer proportion fixed, a 5:1 proportion was used during the rest of the optimization and evaluation process.

For the number of context words, the values $\{0, 1, 2, 3, 4, 10, 15, 20\}$ were tested, yielding only small differences. However, the best-performing model, using only 1 context word on either side of the mention, was significantly better than the default of 5 context words.

For the number of context mentions, the default value of 2 turned out to be optimal, as it worked significantly better than most values from the set $\{0, 1, 3, 4, 5, 6\}$.

Of all parameters, the choice for a set of word embeddings was the most influential. Different sets of GloVe embeddings were tested, varying in dimensionality and number of tokens trained on. The default set was 50D/6B, i.e. 50-dimensional embeddings trained on 6 billion tokens of training data. The sets $\{100D/6B, 200D/6B, 300D/6B, 300D/42B, 300D/840B\}$ were evaluated. All out-performed the default set, and the 300D/42B set performed the best.

Test set	Acc.	Singleton Detection		
		P	R	F1
11-dev	76.37	75.72	90.32	82.38
11-test	77.47	77.26	88.42	82.46
12-dev	79.57	79.77	85.83	82.69
12-test	80.08	81.57	83.78	82.66
12-dev-dM15	79.0	81.1	80.8	80.9
12-dev-BL	68.19	66.90	87.21	75.72

Table 1: Singleton detection performance using the best-scoring model. The CoNLL-2012 training set was used as training data. ‘dM15’ marks the results by de Marneffe et al. (2015) ‘BL’ marks the baseline performance.

3.2 Singleton Detection Results

The final singleton detection model was evaluated on the CoNLL-2011 development and test set, and the CoNLL-2012 test set, in order to evaluate generalization. Results are reported in Table 1. Generally, performance holds up well across data sets, although the results on the 2011 sets are slightly lower than on the 2012 datasets.

At 76-80% accuracy, the multi-layer perceptron is clearly better than the baseline. Performance is also compared to that of the state-of-the-art, by de Marneffe et al. (2015), who only report scores on the CoNLL-2012 development set. The accuracy of my system is 0.6 percentage points higher.

Because word embeddings are the only source of information used by the system, its performance may be vulnerable to the presence of ‘unknown words’, i.e. words for which there is no embedding. Looking at the 2012 development set, we see that classification accuracy for mentions containing one or more unknown words is 76.55%, as compared to 79.63% for mentions without unknown words. The difference is smaller when looking at the context: accuracy for mentions with one or more unknown words in their context is 78.73%, whereas it is 79.79% for mentions with fully known contexts.

3.3 Coreference Resolution Results

Table 2 shows the performance of the Stanford system. Multiple variables governing singleton filtering were explored. ‘NE’ indicates whether named entities were excluded from filtering or not. ‘Pairs’ indicates whether individual mentions are filtered out, or only links between pairs of

mentions. ‘Threshold’ indicates the threshold under which mentions are classified singleton. The threshold value of 0.15 is chosen so that the singleton classification has a precision of approximately 90%

We cannot compare directly to the system of de Marneffe et al. (2015), because they used an older, faulty version of the CoNLL-scorer. For the Stanford system, we therefore compare to a precursor of their system, by Recasens et al. (2013), whose singleton detection system is integrated with the Stanford system. For the Berkeley system, this is not possible. In both cases, we also compare to the system without any singleton detection. Differences were tested for significance using a paired-bootstrap re-sampling test (Koehn, 2004) over documents, 10000 times.

The performance of the different filtering methods is as expected. For more widely applicable filters, precision goes up more, but recall also drops more. For more selective filters, the drop in recall is smaller, but so is the gain in precision. The best balance here is yielded by the ‘Incl./Yes/0.15’-model, the most restrictive model, except for that it includes named entities in filtering. This model yields a small improvement of 0.7 percentage points over the baseline. This is slightly more than the Recasens et al. (2013) system, and also slightly larger than the 0.5 percentage point gain reported by de Marneffe et al. (2015)

NE	Pairs	Threshold	CoNLL-F1
Incl.	No	0.5	50.41*
Incl.	No	0.15	56.73
Incl.	Yes	0.5	55.46*
Incl.	Yes	0.15	57.17*
Excl.	No	0.5	53.64*
Excl.	No	0.15	56.71
Excl.	Yes	0.5	55.96*
Excl.	Yes	0.15	56.92*
Recasens et al. (2013)			56.90*
No Singleton Detection			56.44

Table 2: Performance of the Stanford system on the 2012 development set. Significant differences ($p < 0.05$) from the baseline are marked *.

Table 3 shows the performance of the Berkeley system. Here, singleton detection probabilities are incorporated as a feature. Again, there are

multiple variations: ‘Prob’ indicates each mention was assigned its predicted probability as a feature. ‘Mentions’ indicates each mention was assigned a boolean feature indicating whether it was likely singleton ($P < 0.15$), and a feature indicating whether it was likely coreferential ($P > 0.8$). ‘Pairs’ indicates the same as ‘Mentions’, but for pairs of mentions, where both have $P < 0.15$ or $P > 0.8$. ‘Both’ indicates that both ‘Mentions’- and ‘Pairs’-features are added.

Here, the performance differences are much smaller, yielding only a non-significant 0.3 percentage point increase over the baseline. All models show an increase in precision, and a drop in recall. In contrast, de Marneffe et al. (2015) report a larger performance increase of almost 2 percentage points for the Berkeley system.

Model	CoNLL-F1
Prob	61.83
Prob + Mentions	61.81
Prob + Pairs	62.02
Prob + Both	62.02
No Singleton Detection	61.71

Table 3: Performance of the Berkeley system on the 2012 development set. As a baseline system, the system using the ‘FINAL’ feature set was used. Significant differences ($p < 0.05$) from the baseline are marked *.

4 Discussion

The singleton detection model was optimized with regard to four variables: hidden layer size, number of context tokens, number of context mentions, and set of word embeddings.

For hidden layer size, no clear effect was found. Regarding the set of word embeddings, we found that higher-dimensional embeddings provide better performance, which is in accordance with what Pennington et al. (2014) found. They, and Collobert et al. (2011), also found that embeddings trained on more text performed better on a range of tasks, but we do not see that clearly, here.

As far as the number of context mentions is concerned, the effect is small, and 2 mentions on either side seems an optimal number. Since the closest mentions are likely the most relevant, this makes sense. Also, since the dataset contains both short pronoun mentions and longer NP mentions,

the optimal number is likely a compromise; for pronouns like *it*, one would expect mentions in the left-context to be most important, while this is not the case for NP mentions.

The most counter-intuitive result of parameter optimization is the fact that just 1 context token on either side of the mention proved to be optimal. This contrasts with previous work: de Marneffe et al. (2015) use 2 words around the mention, and semantic information from a larger window, and Bergsma and Yarowsky (2011) use up to 5 words before and 20 words after *it*. Looking at the mention detection literature in general, we see that this pattern holds up: in non-referential *it* detection, larger context windows are used than in works that deal with complete NPs.

Clearly, since large NP mentions already contain more information internally, they require smaller context windows. Likely, the same dynamic is at play here. The OntoNotes dataset contains a majority of NP mentions, and has relatively long mentions, since it only annotates the largest NP of a set of nested head-sharing NPs.

The other main observation to be made on the results is the discrepancy in the effect of singleton information on the Berkeley coreference resolution system in this work and that by de Marneffe et al. (2015). Although singleton detection performance and the performance with the Stanford system are similar, there is almost no performance gain with the Berkeley system here.

Using the Berkeley coreference analyser (Kummerfeld and Klein, 2013), the types of errors made by the resolution systems can be analysed. For the Stanford system, we find the same error type patterns as de Marneffe et al. (2015), which matches well with the similar performance gain. For the Berkeley system, the increases in missing entity and missing mention errors are higher, and we do not find the large decrease in divided entity errors that de Marneffe et al. (2015) found. It is difficult to point out the underlying cause for this, due to the learning-based nature of the Berkeley system. Somehow, there is a qualitative difference between the probabilities produced by the two singleton detection systems.

Regarding the question of how to integrate singleton information in coreference resolution systems, the picture is clear. Both here and in de Marneffe et al. (2015), the best way of using the information is with a high-precision filter, and for

pairs of mentions, rather than individual mentions. The only difference is that excluding named entities from filtering was not beneficial here, which might be due to the fact that word embeddings also cover names, which improves handling of them by the singleton detection model.

For future work, several avenues of exploration are available. The first is to split singleton detection according to mention type (similar to Hoste and Daelemans (2005) for coreference resolution). Since the current model covers all types of mentions, it cannot exploit specific properties of these mention types. Training separate systems, for example for pronouns and NPs, might boost performance.

Another improvement lies with the way mentions are represented. Here, a recursive autoencoder was used to generate fixed size representations for variable-length mentions. However, a lot of information is lost in this compression step, and perhaps it is not the best compression method. Alternative neural network architectures, such as recurrent neural networks, convolutional neural networks, and long short-term memories might yield better results.

In addition, an improved treatment of unknown words could boost performance, since their presence hurts classification accuracy. Currently, an average of all embeddings is used to represent unknown words, but more advanced approaches are possible, e.g. by using part-of-speech information.

To further investigate the interaction between singleton detection and coreference resolution, it would be insightful to look into combining the current system with more recent coreference resolution systems (e.g. Wiseman et al., 2016; Clark and Manning, 2015) which perform better than the Stanford and Berkeley systems. On the one hand, singleton detection information could yield larger gains with these systems, as they might be able to exploit the information better. For example, improved clustering algorithms might benefit more from a reduced number of mentions in the search space. On the other hand, improvements in these systems could overlap with the gain from singleton detection information, lowering the added value of a separate singleton detection system.

All in all, it is shown that a word embedding and neural network based singleton detection system can perform as well as a learner based on hand-

crafted, linguistic-intuition-based features. With a straightforward neural network architecture, and off-the-shelf word embeddings, neither of which is specifically geared towards this task, state-of-the-art performance can be achieved. As an added benefit, this approach can easily be extended to any other language, if word embeddings are available.

Acknowledgements

I am grateful to Jennifer Spender for providing inspiration and feedback during all stages of this project, and to the anonymous reviewers for their kind and useful comments.

References

- Shane Bergsma and David Yarowsky. 2011. NADA: a robust system for non-referential pronoun detection. In Iris Hendrickx, Sobha Lalitha Devi, António Branco, and Ruslan Mitkov, editors, *Anaphora Processing and Applications*, pages 12–23. Springer, Berlin.
- Adriane Boyd, Whitney Gegg-Harrison, and Donna K. Byron. 2005. Identifying non-referential *it*: a machine learning approach incorporating linguistically motivated patterns. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in NLP*, pages 40–47.
- Donna K. Byron and Whitney Gegg-Harrison. 2004. Eliminating non-referring noun phrases from coreference resolution. In *Proceedings of DAARC 2004*, pages 21–26.
- Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Proceedings of ACL 2015*, pages 1405–1415.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.
- Marie-Catherine de Marneffe, Marta Recasens, and Christopher Potts. 2015. Modeling the lifespan of discourse entities with application to coreference resolution. *Journal of Artificial Intelligence Research*, 52:445–475.
- Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of EMNLP 2013*, pages 1971–1982.
- Veronique Hoste and Walter Daelemans. 2005. Learning Dutch coreference resolution. In *Proceedings of CLIN 2004*, pages 133–148.

- Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP 2004*, pages 388–395.
- Jonathan K. Kummerfeld and Dan Klein. 2013. Error-driven analysis of challenges in coreference resolution. In *Proceedings of EMNLP 2013*, pages 265–277.
- Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.
- Chao Ma, Janardhan Rao Doppa, J. Walker Orr, Prashanth Mannem, Xiaoli Fern, Tom Dietterich, and Prasad Tadepalli. 2014. Prune-and-Score: Learning for greedy coreference resolution. In *Proceedings of EMNLP 2014*, pages 2115–2126.
- Vincent Ng and Claire Cardie. 2002. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of COLING 2002*, pages 1–7.
- C. D. Paice and G. D. Husk. 1987. Towards the automatic recognition of anaphoric features in English text: the impersonal pronoun “it”. *Computer Speech and Language*, 2:109–132.
- Haoruo Peng, Kai-Wei Chang, and Dan Roth. 2015. A joint framework for coreference resolution and mention head detection. In *Proceedings of CoNLL 2015*, pages 12–21.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP 2014*, pages 1532–1543.
- Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 shared task: modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on CoNLL*, pages 1–27.
- Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Joint Conference on EMNLP and CoNLL*, pages 1–40.
- Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. 2013. The life and death of discourse entities: Identifying singleton mentions. In *Proceedings of NAACL-HLT 2013*, pages 627–633.
- Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Proceedings of NIPS 2011*, pages 801–809.
- Olga Uryupina. 2003. High-precision identification of discourse new and unique noun phrases. In *Proceedings of the ACL 2003 Student Research Workshop*, pages 80–86.
- Olga Uryupina. 2009. Detecting anaphoricity and antecedenthood for coreference resolution. *Procesamiento del Lenguaje Natural*, 42:113–120.
- Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2011. OntoNotes Release 4.0. DVD.
- Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0. Web Download.
- Sam Wiseman, Alexander M. Rush, Stuart M. Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of ACL 2015*, pages 1416–1426.
- Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. *arXiv preprint arXiv:1604.03035*.
- Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of COLING 2000*, pages 947–953.