Accelerated High-Quality Mutual-Information Based Word Clustering

Manuel R. Ciosici[$€£], Ira Assent[€], Leon Derczynski[£]
[$]UNSILO A/S, [€]Aarhus University, [£]IT University of Copenhagen
{manuel, ira}@cs.au.dk, leod@itu.dk

Abstract

Word clustering groups words that exhibit similar properties. One popular method for this is Brown clustering, which uses short-range distributional information to construct clusters. Specifically, this is a hard hierarchical clustering with a fixed-width beam that employs bi-grams and greedily minimizes global mutual information loss. The result is word clusters that tend to outperform or complement other word representations, especially when constrained by small datasets. However, Brown clustering has high computational complexity and does not lend itself to parallel computation. This, together with the lack of efficient implementations, limits their applicability in NLP. We present efficient implementations of Brown clustering and the alternative Exchange clustering as well as a number of methods to accelerate the computation of both hierarchical and flat clusters. We show empirically that clusters obtained with the accelerated method match the performance of clusters computed using the original methods.

Keywords: word clusters, word representations, efficient computation

1. Introduction

Word clusters have been successfully used in NLP tasks over the past decades, contributing to advances especially in machine translation (Brown et al., 1993; Auli et al., 2013), named entity recognition (Ratinov and Roth, 2009; Ritter et al., 2011), parsing (Koo et al., 2008; Kong et al., 2014), and processing noisy text (Owoputi et al., 2012). They remain a competitive representation useful for many tasks (Dyer et al., 2016; Choi, 2016; Lukasik et al., 2015), yielding superior extrinsic performance in particular when limited data is available (Qu et al., 2015) – which is the case for the majority of languages. More recently, word clusters induced using the Brown clustering algorithm have been shown to be highly effective in encoding syntactic information (Ciosici et al., 2019) and have been used for unsupervised Part-of-Speech tagging (Cardenas et al., 2019).

Brown clustering (Brown et al., 1992) is a commonly used word clustering algorithm, performing a bottom-up, windowed, hard hierarchical clustering based on the global objective of maximized Average Mutual Information (AMI), which is equivalent to the Maximum Likelihood Estimate (MLE) of the underlying language model (Martin et al., 1998). The tension between local merges and the global optimization goal makes the algorithm parallelize hard. Simultaneously, the number of merges considered at any one time directly affects the quality of the final word clustering (Derczynski and Chester, 2016).

While Brown clustering often provides a useful grouping of items into classes based on their distributionality (Ciosici et al., 2019), we find that this process is both slow and also finds poorer optima than using the alternative Exchange clustering. Exchange clustering is an alternative to Brown clustering that optimizes AMI iteratively. Unlike Brown clustering, Exchange clustering outputs a flat clustering. For brevity, we refer to the methods by just *Brown* and *Exchange*, respectively, in the following.

The use of Brown clusters has, from the beginning, been limited by the slow computation time of Brown, even when using a windowed approximation (Brown et al., 1992), as well as lack of fast implementations for the community to use. This is partially due to the use of a global rather than local metric as the objective agglomerative clustering function. In fact, the only available open-source implementation of Brown is over 15 years old (Liang, 2005) and does not include an implementation of Exchange as per the original paper. Most research aimed at runtime performance of Brown so far is limited to relaxations of the underlying language model in the interest of speed (Dehdari et al., 2016; Stratos et al., 2014; Uszkoreit and Brants, 2008).

In this paper, we demonstrate that Brown and Exchange can be combined to speed up computation of Brown clusters while yielding similar quality clusters, in less time, and retaining the tree-based features of the generally slower and more involved Brown algorithm. An added advantage is the ability to move away from the local maxima Brown's greedy algorithm is prone to. We further retro-fit Exchange with stochastic merging, to allow escape from local maxima in Exchange. We contribute our code to the NLP community as the only fast implementation of both Brown and Exchange. It is written in modern C++, and allows hybrid clustering using the two algorithms.[1]

2. Background

We cover Brown and Exchange from an algorithmic point of view, describing their behavior and how they interact with their objective, in order to provide an informed evaluation later. Both are greedy algorithms that construct clusters as a byproduct of training a two-sided class-based language model on a corpus of unstructured text. They model language using a first-order Markov Model where the class c_i (i.e., the cluster) of the predicted word type w_i is conditioned on the class c_{i-1} of the previous word type w_{i-1}, i.e., $P(w_i|w_{i-1}) = P(w_i|c_i)P(c_i|c_{i-1})$ (Brown et al., 1992; Kneser and Ney, 1993; Martin et al., 1998). The optimization goal is to maximize the so-called Average Mutual Information (AMI), a global objective function that acts as

[1] https://github.com/manuelciosici/ExchangeAndBrown

an intrinsic performance measure (equivalent to maximising the likelihood of the underlying class-based model).

2.1. Brown Clustering

During clustering, class pairs are repeatedly merged, with each merge being the one that reduces AMI the least. The initial state is each word type having its own class, and the final state is with all word types in one single class. Each merge alters class-to-class mutual information. The use of AMI, a global objective function with no local distance measures, makes this slow to compute and difficult to parallelize. To reduce computation times, often only a subset of merges are considered. The original paper uses the top 1 000 classes for this purpose (Brown et al., 1992). This number was later abstracted to a variable c. The process is akin to doing beam search for optimal merges, where c determines the beam width.

Later, this was formalized as Generalized Brown (Derczynski and Chester, 2016), which also includes a method for decoupling cluster generation from the search for merge candidates, based on the intuition that each run creates many intermediate clusterings of size 1 to the size of the vocabulary $|V|$. So, to re-create a clustering one can simply re-run the desired number of merges. This then re-purposes beam width as a clustering quality factor, with higher beam widths giving better quality clusterings at the cost of time, running in $O((|V|-c)c^2 + c\log c)$. Setting the beam width c to match the number of words in the vocabulary $|V|$, results in the original, non-windowed Brown clustering.

Brown clustering's intrinsic metric, AMI, demonstrates a typical pattern throughout a clustering run over language data. When clustering is performed with a subset of global states being considered at every step, e.g. running `wcluster` (Liang, 2005) with $c < |V|$, AMI rises monotonically as a greater proportion of the vocabulary is represented in the classes built so far (directly observable in the generalized form's merge file output). AMI peaks at the first point where all words are present in the set of classes being considered, and subsequently decreases during tree building phase. The point of peak AMI is reached when the underlying class-based language model is first fully derived and has the highest performance in modeling the corpus used for training. Crucially, peak AMI is thus a core intrinsic measure of language model quality (Derczynski and Chester, 2016). We use peak AMI to measure a clustering run's performance, i.e., the MLE of the language model.

2.2. Exchange Clustering

Exchange seeks to generate a hard, flat clustering of words from a corpus while optimizing AMI. Like Brown and first-order Markov models, it uses a two-sided model $P(w_i|w_{i-1}) = P(w_i|c_i)P(c_i|c_{i-1})$, where the class c_i of the predicted word w_i is conditioned on the class c_{i-1} of the previous word w_{i-1} (Kneser and Ney, 1993). The target number of clusters is pre-specified and clusters are initialized according to some heuristic. For each iteration, words are examined in order (e.g. in descending corpus frequency), and each word is moved to the cluster which gives the highest AMI (see Algorithm 1). This runs in $O(c|V|n)$, where n is the number of iterations. Exchange moves words from one cluster to another with the aim of minimizing the perplexity of the underlying two-sided class-based bi-gram language model on the provided corpus. Stopping criteria are not explicitly defined for Exchange; in Section 5.3. we present two stopping criteria implemented in our code.

Algorithm 1 Exchange clustering

```
Initialise c empty clusters as C
Assign every word w in vocabulary V from corpus D to a cluster
Iteration count i = 0
while Stopping condition not met do
    for word w_i do
        Calculate best target cluster C_t for w_i
        Move w_i to C_t
    Increment i
```

Several versions of Exchange exist that relax the two-sided class-based language model to a single-sided one, i.e., $P(w_i|w_{i-1}) = P(w_i|c_i)P(c_i|w_{i-1})$ (Goodman, 2001). The relaxation allows for some optimizations via caching, but at the cost of increasing the number of parameters that require estimation by a factor of $|V|$ (Dehdari et al., 2016; Uszkoreit and Brants, 2008).

3. Hybrid Clustering

We introduce a new approach that combines the strengths of the two above algorithms to achieve better runtime performance and better quality at the same time. The core idea is to think of Exchange as an efficient way of generating intial clusters, and of Brown as a refinement method for these clusters, as well as for creating a hierarchy over these clusters.

We propose a hybrid clustering, where one first runs Exchange over the data to initialize a set number of clusters c. We then run Brown clustering to generate a hierarchy, taking as input the clusters C computed by Exchange and also the source corpus.

The original Brown algorithm (Brown et al., 1992) executes in two stages – clustering and tree-building. The point where it switches from the clustering stage to the tree building one is the point of peak AMI (i.e., the point where the language model can model the entire vocabulary).

In hybrid clustering, we use Exchange for the clustering part and Brown for building the hierarchical structure. Thus, for a hybrid clustering, one defines a set number of classes c, runs Exchange over the corpus to compute the c classes, and then runs Brown over the flat clustering.

This has the advantage of giving a second chance to avoid local maxima that the greedy Brown algorithm is likely to corner itself into. For this we use the same frequency-based initialization method as Exchange (Martin et al., 1998).

Hybrid clustering is a faster method, with fewer hyperparameters (no need to specify a window size), that achieves higher AMI, and matches the performance of Brown clusters in downstream tasks.

4. Experimental Setup

We perform experiments to demonstrate the performance gains of our proposed hybrid clustering method with tra-

	Peak AMI	Time (s)	c	speedup
Brown	0.625	9.99	10	2.49×
Hybrid	**0.708**	**4.00**		
Brown	1.188	10.93	40	1.69×
Hybrid	**1.322**	**6.44**		
Brown	1.803	20.19	160	2.73×
Hybrid	**1.845**	**7.38**		
Brown	2.108	49.05	320	6.64×
Hybrid	**2.137**	**7.38**		
Brown	2.504	154.70	640	1.82×
Hybrid	**2.521**	**84.71**		
Brown	2.961	678.80	1280	2.24×
Hybrid	**2.972**	**302.1**		

Table 1: Peak AMI value with varied beam width / cluster count c. Time is wallclock seconds.

ditional Brown using the implementation of Liang (2005). This is run with min-occur set to 1, in order to preserve information and provide a fair, higher-AMI clustering that incorporates a maximum amount of corpus knowledge. In other words, we do not eliminate low frequency words from the vocabulary. Comparing Brown, Exchange, or the hybrid method with other word representations is out of scope for this work. For comparisons of Brown or Exchange with other word representation methods, in particular word vectors, see Bansal et al. (2014), and Qu et al. (2015); for characterization of syntactic information encoded by Brown clusters see Ciosici et al. (2019).

5. Results and Analysis

5.1. Computation and AMI performance

We measure the reduction in computation time and improvement in language modeling. For this first experiment we focus on English, using one million words from Reuters Corpus (RCV) (Rose et al., 2002). We use a machine with dual Intel 8176 and 512GB RAM. Runtime results (mean of three runs) are presented in Table 1.

Our hybrid method provides a higher peak AMI and much lower run-time than Brown, in every case. The gap in AMI closes as the number of clusters generated rises. This is expected; the information present in a clustering rises as the number of clusters increases from 1, though drops as the number of clusters approaches the number of items (i.e. $|V|$, which is typically large for NLP applications). This behavior is preserved in larger corpora, see Table 2 for the full RCV corpus of 114M tokens.

5.2. Downstream performance

As higher AMI can result from overfitting of the language model to the training corpus, we supplement the experiment with extrinsic downstream evaluation in Named Entity Recognition (NER). The aim is to compare the performance of Brown clusters with that of hybrid clusters in a downstream task to study the information encoded.

We perform NER using a classifier based on Conditional Random Fields (CRF) (Derczynski et al., 2015a) with per token cluster bitstring IDs from the cluster hierarchy and character skip-2-gram features over the English newswire train and test splits of the CoNLL 2003 shared task (Tjong Kim Sang and De Meulder, 2003). We choose this simple setup in order to focus on the performance impact of the clustering on the task and to verify cluster quality.

$c=$	10	40	160	640	2560
Brown	0.565	1.143	1.671	2.154	2.631
Hybrid	**0.694**	**1.271**	**1.747**	**2.189**	**2.650**

Table 2: Peak AMI for Exchange and Brown over the complete RCV1 dataset, varying c; $i = 10$.

$c=$	10	40	160	320	640	2560
Brown	**72.86**	73.41	73.99	74.72	74.60	**74.32**
Hybrid	72.44	**74.04**	**74.08**	**74.98**	**75.35**	74.16

Table 3: Extrinsic F1 on CoNLL 2003 NER.

To avoid further AMI loss, instead of "shearing" clusters at fixed bit depths, we use roll-up feature generation (Derczynski and Chester, 2016). Brown cluster trees are asymmetrical, so when one derives Brown clustering features by truncating bit strings (e.g. cutting them at length 4, 6, 10 and 20 as per Ratinov and Roth (2009)), the result is often a "false" clustering that never actually occurred during the running of Brown. Such truncation-based extractions offer lower AMI and yield reduced extrinsic performance (Derczynski and Chester, 2016). To retain the maximum AMI, then, one may instead trivially "re-play" the clustering up to the point where the number of leaves on the tree matches the desired number of classes, preserving the structure intended in the algorithm's output. This has the added advantage of allowing any arbitrary number of output clusters o both up to and beyond c all the way to $|V|$, instead of being constrained to $o \leq c; o \leq 2^n; n \in \mathbb{Z}^+$ as the older "shearing" method is.

Extrinsic $F1$ results in Table 3 show that our hybrid approach's performance matches that of Brown, indicating no harmful overfitting, and demonstrating that the speed-up is not at the expense of cluster quality.

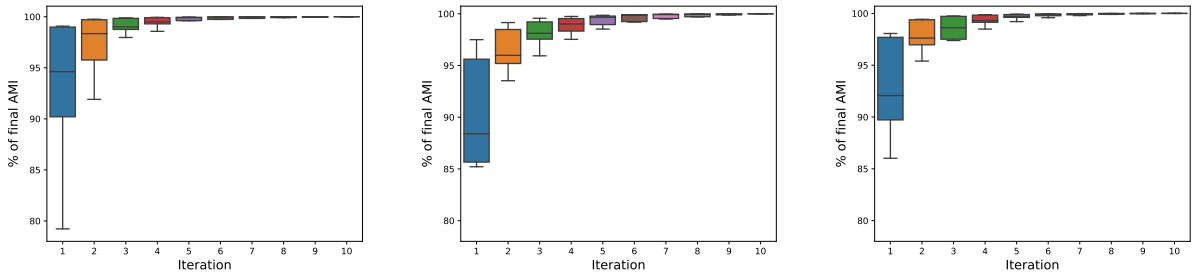
5.3. Stopping criteria

The point at which to stop Exchange is not well-defined in its original presentation (Kneser and Ney, 1993). We propose three simple stopping criteria.

5.3.1. AMI Threshold

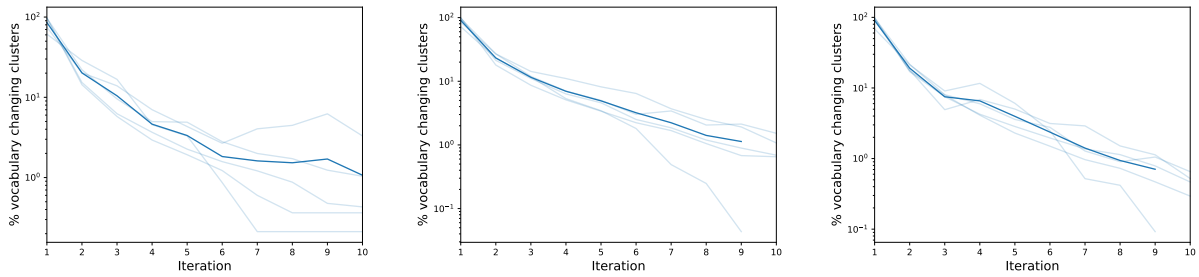
The first is a cut-off for gradient over iterations. If AMI increase dips below a given threshold between iterations, the algorithm stops. This bound can be specified regardless of the input data size (which drives the AMI values).

The box plots in Figure 1 show the percentage of final AMI achieved in each of the first 10 iterations of Exchange, measured over 5 runs, with different values for the desired number of clusters (18, 50, 100, 500, 800) according to best practices (Derczynski et al., 2015b), using the English, French, and Czech Universal Dependencies 2.1 as input data (Leung et al., 2017). The vast majority of the final AMI is achieved within three or four iterations. Thus, stopping Exchange after three iterations results in a three-fold speedup compared to suggestions in the literature (Uszkoreit and Brants, 2008; Martin et al., 1998). This stopping criterion can easily be used in our hybrid method as well.



(a) English UD (b) Czech UD (c) French UD

Figure 1: Per-iteration Exchange AMI as final AMI percentage. Whiskers mark maximum and minimum values.



(a) English UD (b) Czech UD (c) French UD

Figure 2: Number of words that participate in swaps as the iterations progress. Individual runs are shown in pale blue, and the mean in a stronger tone.

5.3.2. Words-moved Threshold

We see that AMI approaches its peak quickly and progresses asymptotically to some theoretical peak. Indeed, little is left to be gained after the first few iterations. This motivates the AMI Threshold criterion (Figure 1). However, this is based on an unknown quantity – the AMI ceiling – which cannot immediately be derived from what one knows about the data before processing.

A more concrete metric is the number of items (words) that swap cluster assignment per iteration. This has a known maximum, $|V|$ and minimum, 0. The swapping behaviour for three languages can be seen in Figure 2, which shows the proportion of words moved per iteration, over multiple runs of Exchange. We see that after the first three iterations, only a small proportion of the vocabulary, less than 10%, continue to swap clusters. Candid examinations indicate that this 10% are typically not common words, i.e. they represent a small part of the overall frequency mass of the corpus' word instances.

Therefore, the next stopping criterion is based on the number of words that move in an iteration: if this falls below threshold m, clustering stops. Observations of the Exchange clustering process suggested that many words will find an optimal cluster regardless of the distribution of most other words. We attribute this to the Zipf-Mandelbrot distribution of word frequencies (Montemurro, 2001) and the effect frequency has on each word type's contribution to global mutual information. That is, the majority of words share little mutual information with most other words; thus, most words may be clustered quickly and will not later be moved – later cluster changes are unlikely to have strong far-reaching effects. This motivated stopping after a certain number of words have been moved. One may define m, the words-moved threshold, based on e.g. a fraction of the vocabulary size $|V|$, or on a total amount of frequency mass in the data and the proportion of terms that comprise it.

c	Speedup	AMI	$i = 10$ AMI	Brown AMI
10	9.29×	0.358	50.5%	57.2%
40	8.01×	0.729	55.2%	61.4%
160	6.05×	1.187	64.3%	65.8%
320	5.97×	1.535	71.8%	72.8%
640	7.24×	1.978	78.5%	79.0%
1280	6.97×	2.540	85.5%	85.8%

Table 4: Performance of one-iteration Exchange, vs. 10-iteration, and Brown. Comparisons are made with AMI values in Table 1.

5.3.3. Minimal Iteration Threshold

From the words-moved threshold, one can see the Exchange does most of the important work early. Based on this observation, the third stopping criterion is a radical option: perform one iteration. Exchange is generally faster than Brown, and also a single iteration is meaningful as opposed to Brown, which is agglomerative and so intrinsically requires multiple iterations to reach the target number of clusters (excluding the unusual extreme case $c = |V|-1$. As with many clustering and other machine learning algorithms, depending on the starting conditions, the most radical and highest volume of changes tend occur in the earliest epochs. This is the phenomenon behind for exam-

	English UD		French UD		Czech UD	
c \ r	0	0.75	0	0.75	0	0.75
18	**0.995**	0.992	1.253	**1.307**	0.828	**0.832**
50	1.318	**1.371**	1.768	**1.816**	**1.189**	1.129
100	1.596	**1.628**	2.110	**2.147**	**1.404**	1.402
500	**2.370**	2.362	2.821	**2.824**	1.988	**1.999**
800	**2.689**	2.685	3.056	**3.063**	**2.246**	2.244

Table 5: Peak AMI with stochastic merging. $i = 50$

ple the prevalence of warm-up phases (Goyal et al., 2017), where learning rates are scaled for the explicit purpose of attenuating these large moves. In this situation, we believe there may be an extreme speed advantage in performing just one iteration, which may provide a large number of well-clustered items.

Noting the high proportion of AMI achieved in the first iteration of Exchange (Figure 1), we experiment with single-iteration Exchange as a rapid clustering, running in only $O(c|V|)$. Table 4 shows results of evaluating this one-shot approach in terms of time taken, and the proportion of AMI achieved after a full Brown clustering and of a 10-iteration Exchange clustering. While being fast, the AMI drop is substantial at low cluster counts. At higher cluster counts one-shot Exchange approaches multiple-iteration performance, and remains faster. This one-iteration criterion is thus an option for extreme runtime optimisation at the cost of some AMI, with speedup remaining good at higher values of c while AMI retention increases.

5.4. Stochastic Merging & Model Selection

Exchange is a greedy method, which depends on both the prior state and the order in which words are examined, leaving little opportunity for escaping local maxima. Stochastic swapping with chance r, with cluster assignment distributed uniformly, gives a chance to escape maxima. Stochastic swapping results are given in Table 5. The value of 0.75 is chosen to give a strong perturbation with still some chance for a signal to come through; for a given dataset, it would be best chosen through e.g. Bayesian optimisation. For some values of c this is helpful for English and Czech; in French, we always see an improvement.

6. Conclusion

This paper proposes a method for induction of distributionally-derived hierarchical word clusterings both with improved speed and at high quality by combining Exchange and Brown clustering, versus using Brown alone. Further, it introduces a simple but effective method for avoiding local maxima during Exchange clustering, leading to further performance boosts. C++ code for the hybrid tool is made available with this paper.[1]

Acknowledgements

This work was supported by the MultiStance research project at the IT University of Copenhagen, by Danmarks Frie Forskningsfond through the Verif-AI project, number 9131-00131B, and by Danmarks Innovationsfond Industrial PhD project, number 5016-00116B.

7. References

Auli, M., Galley, M., Quirk, C., and Zweig, G. (2013). Joint language and translation modeling with recurrent neural networks. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1044–1054, Seattle, Washington, USA, October. Association for Computational Linguistics.

Bansal, M., Gimpel, K., and Livescu, K. (2014). Tailoring Continuous Word Representations for Dependency Parsing. In *Acl*, pages 809–815.

Brown, P. F., Della Pietra, V. J., deSouza, P. V., Lai, J. C., and Mercer, R. L. (1992). Class-based *n*-gram models of natural language. *Computational Linguistics*, 18(4):467–480.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Cardenas, R., Lin, Y., Ji, H., and May, J. (2019). A grounded unsupervised universal part-of-speech tagger for low-resource languages. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2428–2439, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Choi, J. D. (2016). Dynamic feature induction: The last gist to the state-of-the-art. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 271–281, San Diego, California, June. Association for Computational Linguistics.

Ciosici, M., Derczynski, L., and Assent, I. (2019). Quantifying the morphosyntactic content of Brown clusters. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1541–1550, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Dehdari, J., Tan, L., and van Genabith, J. (2016). BIRA: Improved predictive exchange word clustering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1169–1174, San Diego, California, June. Association for Computational Linguistics.

Derczynski, L. and Chester, S. (2016). Generalised Brown clustering and roll-up feature generation. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Derczynski, L., Augenstein, I., and Bontcheva, K. (2015a). USFD: Twitter NER with drift compensation and linked data. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 48–53, Beijing, China, July. Association for Computational Linguistics.

Derczynski, L., Chester, S., and Bøgh, K. S. (2015b). Tune your Brown clustering, please. In *International Conference Recent Advances in Natural Language Processing,*

RANLP, volume 2015, pages 110–117. Association for Computational Linguistics.

Dyer, C., Kuncoro, A., Ballesteros, M., and Smith, N. A. (2016). Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California, June. Association for Computational Linguistics.

Goodman, J. T. (2001). A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403 – 434.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. (2017). Accurate, large minibatch SGD: Training ImageNet in 1 hour. Technical report, Facebook.

Kneser, R. and Ney, H. (1993). Improved clustering techniques for class-based statistical language modelling. In *Third European Conference on Speech Communication and Technology*, pages 973–976.

Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., and Smith, N. A. (2014). A dependency parser for tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1001–1012, Doha, Qatar, October. Association for Computational Linguistics.

Koo, T., Carreras, X., and Collins, M. (2008). Simple semi-supervised dependency parsing. In *Proceedings of ACL-08: HLT*, pages 595–603, Columbus, Ohio, June. Association for Computational Linguistics.

Leung, H., Li, C. Y., Li, J., Li, K., Ljubešić, N., Loginova, O., Lyashevskaya, O., Lynn, T., Macketanz, V., Makazhanov, A., et al. (2017). Universal Dependencies 2.1. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Liang, P. (2005). Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

Lukasik, M., Cohn, T., and Bontcheva, K. (2015). Classifying tweet level judgements of rumours in social media. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2590–2595, Lisbon, Portugal, September. Association for Computational Linguistics.

Martin, S., Liermann, J., and Ney, H. (1998). Algorithms for bigram and trigram word clustering. *Speech Communication*, 24(1):19 – 37.

Montemurro, M. A. (2001). Beyond the Zipf–Mandelbrot law in quantitative linguistics. *Physica A: Statistical Mechanics and its Applications*, 300(3):567 – 578.

Owoputi, O., Dyer, C., Gimpel, K., and Schneider, N. (2012). Part-of-speech tagging for Twitter: Word clusters and other advances. Technical report, School of Computer Science Carnegie Mellon University.

Qu, L., Ferraro, G., Zhou, L., Hou, W., Schneider, N., and Baldwin, T. (2015). Big data small data, in domain out-of domain, known word unknown word: The impact of word representations on sequence labelling tasks. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 83–93, Beijing, China, July. Association for Computational Linguistics.

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado, June. Association for Computational Linguistics.

Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011). Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Rose, T., Stevenson, M., and Whitehead, M. (2002). The Reuters corpus volume 1 -from yesterday's news to tomorrow's language resources. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain, May. European Language Resources Association (ELRA).

Stratos, K., Kim, D.-k., Collins, M., and Hsu, D. J. (2014). A spectral algorithm for learning class-based n-gram models of natural language. In *Proc. UAI*, pages 762–771.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Uszkoreit, J. and Brants, T. (2008). Distributed word clustering for large scale class-based language modeling in machine translation. In *Proceedings of ACL-08: HLT*, pages 755–762, Columbus, Ohio, June. Association for Computational Linguistics.