From neighborhood to parenthood: the advantages of dependency representation over bigrams in Brown clustering

Simon Šuster
University of Groningen
Netherlands
s.suster@rug.nl

Gertjan van Noord
University of Groningen
Netherlands
g.j.m.van.noord@rug.nl

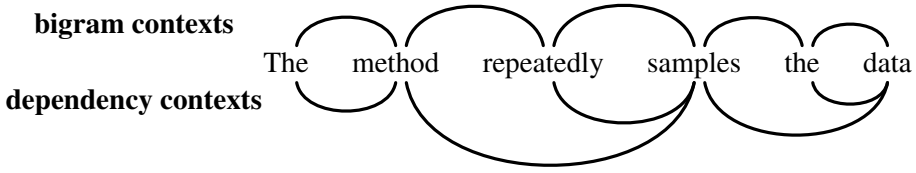
Abstract

We present an effective modification of the popular Brown et al. 1992 word clustering algorithm, using a dependency language model. By leveraging syntax-based context, resulting clusters are better when evaluated against a wordnet for Dutch. The improvements are stable across parameters such as number of clusters, minimum frequency and granularity. Further refinement is possible through dependency relation selection. Our approach achieves a desired clustering quality with less data, resulting in a decrease in cluster creation times.

1 Introduction

Semi-supervised approaches have been successful in various areas of natural language processing. Among a plethora of clustering techniques, Brown clustering (Brown et al., 1992) is popular for its conceptual simplicity, available implementations (Liang, 2005; Stolcke, 2002), and because the resulting word clusters can be helpful for several tasks. Clusters are used as syntactic and semantic *generalizations* of words, requiring fewer model parameters.

Brown clustering (section 2) groups words based on shared context. However, only immediately adjacent words are taken into account as recognized e.g. by Koo et al. (2008), Sagae and Gordon (2009), and Grave et al. (2013). For example, even though verbs constitute an informative context for object nouns, they are rarely considered in Brown clustering, unlike in dependency-based clustering. The difference between the contexts can be illustrated with the following example:



The bigram context thus fails to capture the relation between the object *data* and the predicate *samples*, as well as the one between the subject *method* and the predicate. Furthermore, the dependency representation rightly ignores some of the less informative contexts coming from immediately adjacent words. For example, there is no relation between the predicate *samples* and the article *the* to the right.

It might be preferable therefore to induce word clusters based on the dependency relations in which the words occur. In section 3, we present how this relates to Brown clustering, and we modify the code by Percy Liang, so that dependency clustering can be used. We evaluate clusters in a wordnet-based similarity experiment. Dependency clustering yields superior clusters for Dutch across different settings of parameters such as number of clusters, frequency threshold and level of granularity. Selecting specific dependency relation labels and using data obtained from them as input to clustering further improves the clustering quality. The proposed adaptation of Brown clustering does not change the complexity of the

algorithm, and—although we assume that syntactically parsed text is available—it requires much less data for a desired level of clustering quality.

2 The Brown clustering algorithm

Brown clustering (Brown et al., 1992) is an agglomerative algorithm that induces a hierarchical clustering of words. It takes a tokenized corpus and groups words into k clusters identified by bit strings, representing paths in the induced binary tree in which the leaves are word clusters. Prefixes of the paths can be used to achieve clusters of coarser granularity (Sun et al., 2011; Turian et al., 2010). The obtained clusters contain words that are semantically related, or are paradigmatic or orthographic variants.[1]

The algorithm starts by putting k most frequent words into distinct clusters. Then, the k+1$^{\text{th}}$ most frequent word is assigned to a new cluster, and two among the resulting k+1 clusters are merged, i.e. the pair that maximizes the average mutual information of the current clustering. This process is repeated until all words have been merged. The resulting k clusters are then merged to build the binary tree. The version of the algorithm optimized for speed runs in $O(k^2|\mathcal{V}|)$, with $|\mathcal{V}|$ the vocabulary size.

Brown clustering has been used extensively in supervised NLP tasks such as parsing (Koo et al., 2008; Candito and Crabbé, 2009; Haffari et al., 2011), named-entity recognition (NER) and chunking (Turian et al., 2010), sentiment analysis (Popat et al., 2013), relation extraction (Plank and Moschitti, 2013), unsupervised semantic role labeling (Titov and Klementiev, 2012), question answering (Momtazi et al., 2010), POS tagging (Owoputi et al., 2013) and speech recognition with recursive neural networks (Shi et al., 2013). Recently, multilingual clustering has also been proposed (Täckström et al., 2012; Faruqui and Dyer, 2013).

Among the most frequently recognized limitations (cf. Koo et al. (2008); Chrupala (2011)) are a) the hard nature of the clustering, b) relatively long running time[2] and c) insensitivity to wider context. Our method attempts to overcome the final disadvantage. As it requires less data, it also reduces the running time.

Leveraging syntactic context for word representations has been explored, among others, in Lin (1998) on distributional thesauri; Haffari et al. (2011) on combining Brown clusters and word groupings from split non-terminals; Sagae and Gordon (2009) on using unlexicalized syntactic context in hierarchical clustering; Van de Cruys (2010) and Padó and Lapata (2007) on comparison of window- and syntactic-based word space models; and Boyd-Graber and Blei (2008) on syntactic topic models.

The work closest to ours is that of Grave et al. (2013). The authors show that clusters obtained from dependency trees outperform standard Brown clustering when used as features in super-sense tagging and NER. Their focus is on a generalization of Brown clustering with Hidden Markov models (extending Markov chains to trees), allowing the creation of soft clusters.[3] Learning and inference are done with online expectation-maximization and belief propagation.

Whereas Grave et al. focus on new learning methods for clustering with HMMs on dependency trees, we take an in-depth look at parameters and choices that are standardly considered using the (Brown et al., 1992) algorithm. We show that the advantage of dependency clustering can be observed throughout different parametrizations of cluster capacity, granularity level, frequency thresholding and other criteria (section 6), and that the advantage is roughly constant for varying amounts of input data. Finally, we provide new insight in the advantage of selective dependency clustering, in which the data obtained only from specific dependency relations lead to better clusters. Our approach constitutes a straightforward extension of Brown clustering, and only required a simple modification of the Brown clustering code.

[1] We are using the term *semantic relatedness* in its broadest possible scope. Words or clusters are semantically related when they have any kind of semantic relation: synonymy, meronymy, antonymy, hypernymy etc. (Turney and Pantel, 2010).

[2] Although coarser clustering (k<1000) can mean more practical running times, as the clustering depends quadratically on k.

[3] This approach allows to capture homonymy/polysemy, with the idea that when a word representation is needed, it can be obtained in a context-sensitive way (Huang et al., 2011; Nepal and Yates, 2014). This is certainly an important advantage over Brown clustering in which the mapping between a word and a cluster is deterministic; however, it comes with its own disadvantages: creating context-sensitive representations requires (potentially) costly inference; furthermore, HMM-based clustering does not build nor lends itself easily to a hierarchy, which is often exploited during feature creation in supervised learning to control cluster granularity (see the end of section 5.2)

3 Extension of the Brown clustering

The bigram language model underlying Brown clustering takes the probability of a sentence as the product of probabilities of words based on immediately preceding words. In contrast, we replace this by a *dependency* language model (DLM), which defines the probability of a sentence over dependency trees (Shen et al., 2008). This probability can be factorized in different ways (Chen et al., 2012; Charniak, 2001; Popel and Mareček, 2010), but the common idea is that a word is conditioned on some history, where the link between the two is a dependency. In practice, the history can include the immediate parent of the word, which can be either a lexical head or the artificial root node, as well as siblings between the child and the parent. Our take on DLM is similar to Charniak (2001) and Popel and Mareček (2010): the probability of a word is conditioned simply on its parent. This is the same view as taken by Grave et al. (2013).

The Brown clustering objective is to find such a deterministic clustering function $\mathcal{C}$ mapping each word from the vocabulary $\mathcal{V}$ to one of K clusters that maximizes the likelihood of the data. The likelihood of a sequence of word tokens, $\mathbf{w} = \langle w_i \rangle_{i=1}^{m}$, with each $w_i \in \mathcal{V}$, factors as

$$L(\mathbf{w}; \mathcal{C}) = \prod_{i=1}^{m} p(\mathcal{C}(w_i)|\mathcal{C}(w_{i-1}))p(w_i|\mathcal{C}(w_i)), \tag{3.1}$$

where $\mathcal{C}(w_0)$ is a special start-of-sequence symbol. As shown by Brown et al. (1992), by taking the negative logarithm and using the ML estimates, the equation 3.1 is decomposed to the negative entropy of the sequence $\mathbf{w}$ and mutual information between adjacent clusters. Since the entropy is independent of the clustering function, the objective amounts to finding such $\mathcal{C}$ that maximizes the mutual information.

For *dependency clustering*, we change the cluster transition probability so that conditioning is on the cluster of the parent of the word at position i, instead of on the cluster of the previous word:

$$L'(\mathbf{w}; \mathcal{C}) = \prod_{i=1}^{m} p(\mathcal{C}(w_i)|\mathcal{C}(w_{\pi(i)}))p(w_i|\mathcal{C}(w_i)), \tag{3.2}$$

where i ranges over all children in a tree and π is a function from the children to their unique parents (which include the special root of the tree). Calculation of the mutual information changes only to the extent that count tables no longer represent adjacency relationship (bigrams) between words but parenthood (child–parent relation).

4 Evaluation task

We evaluate our word clusters by following the method of Van de Cruys (2010) for evaluating vector space models. The method is based on a wordnet for Dutch and assumes that two semantically related words also occur close to each other in the wordnet hierarchy.[4] We use Cornetto (Vossen et al., 2013), which includes more than 92,000 form-POS pairs described in terms of lexical units, synsets and other criteria. For calculating similarity scores, we treat Cornetto as a digraph, with nodes constituting synsets and arcs constituting hypernymic relations, and adopt the Lin similarity measure (Lin, 1998)[5] in combination with the ontological variant of Information Content[6].

Evaluation is guided by a list of 10,000 most frequent words from SoNaR, a 500M-word reference corpus for Dutch.[7] Every word is compared to other words in the same cluster, and the average similarity for all comparisons is taken as the final score. The described method is well suited for measuring intracluster quality, yet useful information about word similarity is available also by looking at neighboring

[4]For English, several semantic similarity datasets are available (such as WordSimilarity-353 (Finkelstein et al., 2001)), some of which can identify the type of relatedness captured. We are not aware of such datasets for Dutch.

[5]Which is a function of the IC of the least common subsumer of two synsets and the IC of individual synsets. The score ranges between 0 and 1.

[6]Which is the negative logarithm of $(|\mathcal{L}| + 1)^{-1}((|\mathcal{L}_s|/|\mathcal{S}_s|) + 1)$, where $\mathcal{L}$ are the leaves of the hierarchy, $\mathcal{L}_s$ are the leaves reachable from a synset s, and $\mathcal{S}_s$ are the subsumers of s (Sánchez et al., 2011).

[7]`http://lands.let.ru.nl/projects/SoNaR`

clusters in the binary tree. This *inter*cluster quality, according to which clusters that are close in the binary tree are more similar than clusters that are far apart, can be captured indirectly by evaluating using different bit substrings. In this way, when a substring is used, two or more semantically related, but isolated clusters are merged, which should result in a drop in clustering quality (semantic relatedness tends to "dissolve" when merging).

For both standard and dependency Brown clustering, the same set of sentences is used. From SoNaR, we sampled sentences amounting to roughly 46M words, which is comparable to the count for English datasets of Koo et al. (2008) and Turian et al. (2010). The sentence length was restricted to five or more words to exclude noisy text. Corpus annotation was removed.

For dependency clustering, the dataset was lemmatized and parsed with the Alpino parser (Van Noord, 2006), an HPSG parser with a maxent disambiguation component, achieving labeled dependency accuracy of around 90.5 for Dutch.[8] The parsing accuracy is likely to be lower on our dataset, but we expect this effect to be small since Alpino has been shown to be relatively insensitive to domain shifts compared to some entirely data-driven parsers (Plank and van Noord, 2010). For default clustering, we only use first-order dependencies produced by the parser. The bilexical counts (head and dependent regardless of the relation label) serve as input for dependency clustering.

5 Experiments and Results

The main parameter for word clustering is the number of clusters k, which we set to either 1000 or 3200,[9] except when measuring clustering capacity, for which smaller values of k are used. Additionally, we limit the minimum frequency of words in clustering to three, unless stated otherwise. The vocabulary size for k=1000 clustering with applied frequency threshold is around 237,000. We use a paired t-test to check for statistical significance of observed differences in means.

5.1 Cluster examples

In Table 1, we show both the versatility of dependency clusters by dividing the examples in five groups (A–E), and the similarity of clusters within group. The longer the common bit substring between clusters, the closer they are in the hierarchy. Group **A** includes words describing professions or people's roles and functions. Group **B** lists personal pronouns, including reflexive pronouns (B2), where substantial differentiation exists with many singleton clusters. Clusters are capable of grouping orthographic variants (D1; *email* and *e-mail*) and diminutives (*sms_DIM*, corresponding to Dutch *smsje*). Because first and last names are extremely common in our corpus, clustering creates fine-grained distinctions between these (**C**). C1 groups names of presidents, whereas C2 and C3 distinguish between feminine and masculine names. Measurable concepts are included in **E**.

5.2 Cluster quality

Table 2 presents the general quality of standard and dependency clustering. The results for 1000 and 3200 clusters (in the latter we use a higher frequency threshold for faster computation) show that we obtain a higher similarity score for 3200 clusters compared to 1000, and a more marked difference between standard and dependency clustering in the case of k=3200 (Δ=0.019). We also looked at how many words from the frequency list were evaluated successfully. The recall depends on the success of mapping between words and synsets as well as the success of finding the word in one of the clusters. The latter factor influences the recall to a much lesser degree, as almost all words are found in the clustering. For 3200 clusters with the minimum frequency set to fifty, approximately 5000 words are successfully evaluated, whereas for 1000 clusters, this number is around 7000.[10] These numbers are not affected by the type of clustering (standard or dependency).

[8] Strictly speaking, the output of lemmatization is root forms. We perform this preprocessing step to increase the number of times that a word is successfully matched in the wordnet hierarchy and evaluated.

[9] Which are standardly encountered throughout the literature. For k above 3200, the algorithm falls short of practicality on current hardware assuming a single-core implementation.

[10] The difference between the figures occurs because of a different frequency threshold.

Group	Cluster id	Most frequent words	Left
A1	001010001011100	aannemer, huis_arts, bakker, notaris, apotheker, makelaar *contractor*, *family doctor*, *baker*, *lawyer*, *pharmacist*, *estate agent*	+57
A2	001010001011011	analist, criticus, waarnemer, kenner, commentator, mens_recht_organisatie *analyst*, *reviewer*, *observer*, *expert*, *commentator*, *human rights organization*	+8
A3	0010100010111110	ondernemer, zakenman, bedrijf_leider, zelfstandige, koopman, starter *entrepreneur*, *businessman*, *manager*, *self-employed*, *merchant*, *starter*	+18
B1	011101111011110	mij *me*	0
B2	01110111101110	zichzelf, mezelf, jezelf, onszelf, mijzelf, uzelf *him/herself*, *myself*, *yourself*, *ourselves*, *myself*, *yourself*	0
B3	01110111101101	hem *him*	0
B4	01110111101100	hen *them*	0
C1	0011010010	Bush, Obama, Clinton, Poetin, Chirac, Sarkozy *Bush*, *Obama*, *Clinton*, *Putin*, *Chirac*, *Sarkozy*	+95
C2	0011000111010	Sarah, Kim, Nathalie, Justine, Kirsten, Tia, Eline	+12
C3	0011000111011	David, Jimmy, Benjamin, Samuel, Tommy, Sean	+98
D1	001011100010101	email, mail, sms, sms_DIM, e-mail, mail_DIM	+13
D2	001011100010100	telefoon, satelliet, telefonie, telefoon_lijn, Explorer, muziek_speler, iTunes *telephone*, *satellite*, *telephony*, *telephone line*, *Explorer*, *music player*, *iTunes*	+7
E	001000010110101	inkomen, energie_verbruik, minimum_loon, cholesterol, opleidingsniveau, *income*, *energy consumption*, *minimum wage*, *cholesterol*, *level of education*, IQ, alcohol_gehalte *IQ*, *alcohol content*	+32

Table 1: Example dependency clusters obtained from a run with number of clusters set to 3200 and minimum frequency to 50. The underlined part of the bit string indicates the longest common substring within one group. English translation of the Dutch original is given in italics and is left out when clear from the original. Column *Left* indicates the remaining number of (less frequent) words in the cluster.

k	Brown	DepBrown	Δ
1000	0.191	0.196	+.005*
3200	0.279	0.298	+.019**

Table 2: Lin similarity scores for standard *Brown* clustering and dependency Brown clustering (*DepBrown*), with k the number of clusters. Δ=DepBrown – Brown. Frequency threshold of 50 is used for clustering with $k = 3200$. *: statistically significant with $p < 0.05$, **: statistically significant with $p < 0.001$.

Results for four different clustering parametrizations are shown in Table 3. One way of controlling the granularity is to choose the number of output clusters k. As shown in the table under CAP ("capacity"), dependency clustering achieves a better quality regardless of the choice of k, and in general, choosing a smaller k decreases quality, which is compatible with the observations of Turian et al. (2010) in their chunking experiments.

An effect similar to that of controlling capacity can be achieved by making use of the fact that the induced structure is a hierarchy.[11] By choosing a path prefix length that is shorter than the maximum length, we control the cluster granularity (denoted in the table as PREF-*). For different tasks, different path prefixes might be appropriate (Sun et al., 2011; Koo et al., 2008; Miller et al., 2004). For example, one might prefer coarser distinctions (i.e. shorter bit strings) in parsing, while finer granularity might be necessary to obtain effective representations of proper names in NER. We ran the experiment with prefix length ranging from one to eighteen, and show a selection of four settings in the table. Across the board, dependency clustering yields better results than standard clustering. Naturally, with shorter prefixes the quality decreases, which is explained by increasing word population in the clusters, with more and more

[11]The parameter k needs to be chosen before clustering, whereas the hierarchical structure can be exploited during feature preparation based on already existing clusters.

Setting	k	*min*	Brown	DepBrown	Δ
CAP	200	10	0.148	0.157	+.009
	400	10	0.169	0.175	+.006
	600	10	0.182	0.191	+.009
	800	10	0.191	0.205	+.014
PREF-16	1000	10	0.2	0.215	+.015
PREF-12	1000	10	0.187	0.202	+.015
PREF-8	1000	10	0.159	0.168	+.009
PREF-4	1000	10	0.114	0.127	+.013
FREQ	1000	5	0.196	0.204	+.008
	1000	10	0.202	0.216	+.014
	1000	20	0.206	0.221	+.015
	1000	30	0.209	0.224	+.015
	1000	50	0.216	0.227	+.011
NOUNS	1000	3	0.272	0.279	+.007

Table 3: Lin similarity scores for standard *Brown* clustering and dependency Brown clustering (*DepBrown*), with *k* the number of clusters, *min* the minimum frequency of words. CAP: varying k, fixed min; FREQ: varying min, fixed k; NOUNS: evaluating only nouns, PREF-n: size of bit-string prefix, Δ=DepBrown – Brown. All the results reported for *DepBrown* are significantly different from *Brown* with $p < 0.001$.

distant (both hierarchically and semantically) clusters being merged.

By inspecting individual clusters, we observe that frequent words in a cluster exhibit clear semantic relatedness, but that rare words are often semantically quite unrelated.[12] This is confirmed by our results in which the quality of the clustering improves approximately logarithmically with frequency threshold increasing (FREQ). The margin between standard and dependency clustering is also increasing as we increase the threshold. In practice, Brown clusters appear to be equally useful with a high frequency threshold (Owoputi et al., 2013) as without thresholding (Koo et al., 2008; Turian et al., 2010).

We also investigate the quality of nouns only, to facilitate the comparison to Van de Cruys (2010). We observe a considerable gain in quality when only nouns are used compared to using all parts of speech — the Lin score is increased by 0.08. In the noun-only evaluation, dependency clustering achieves a higher score (0.279) than standard clustering (0.272). Van de Cruys (2010) shows that syntactic vector space models outperform window-based models, which is confirmed by our finding for word clustering as well. In his work, syntactic vector space models yield a 0.04 advantage in Lin score, whereas our dependency clusters achieve a less marked advantage, reaching up to 0.019 in Lin score. A possible explanation for this difference is that in his evaluation an average over only five most similar nouns is taken, whereas we impose no such restriction. We would like to point out that our work does not aim to compare and discuss the merits of clustering and vector space models as possible techniques for obtaining word representations, but rather to provide a comprehensive comparison of standard Brown clustering and its dependency extension.

5.3 Learning curves

Figure 5.2 shows the amount of data needed to achieve a certain quality of clustering. For clustering on ten thousand sentences the similarity score is around 0.14, with a higher score for standard clustering. For each subsequent addition of data, dependency clustering outperforms standard clustering. In order to achieve the highest score attained by standard clustering (0.19), resulting from clustering on 2.4 million sentences (41 million words), dependency clustering requires only slightly more than 500 thousand sentences (8.5 million words). This observation is advantageous especially because less data means

[12] Although cf. Turian et al. (2010) who show that Brown clustering has a superior representation for rare words than neural word embeddings in their experiment.

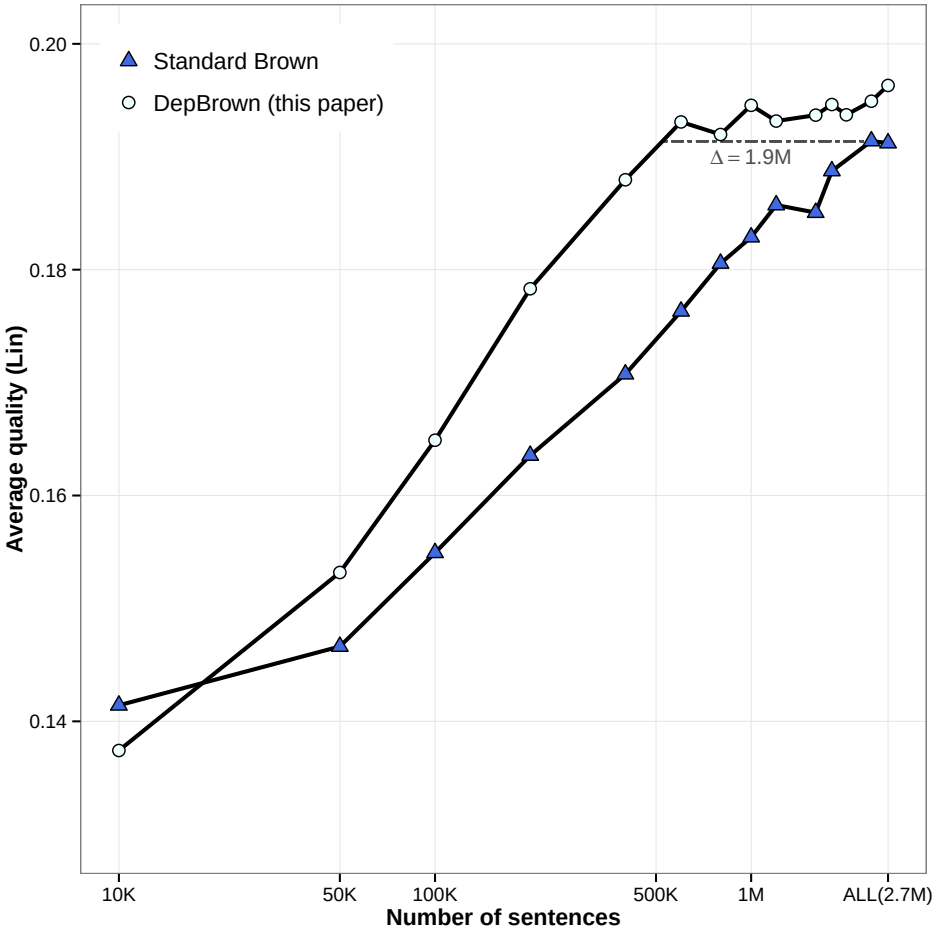


Figure 1: Learning curves for standard and dependency Brown clustering with 1000 clusters and a frequency threshold of 3. Dashed line displays the difference in amount of data needed for *DepBrown* to achieve the best quality of *Brown*. Using all, 2.7 million sentences from the corpus (*ALL*) corresponds to 46 million words.

shorter running time for clustering as the number of word types is reduced.

5.4 Refinement of dependency clusters

Our dependency clustering described in the previous sections operates on words appearing in all dependency relations. We now investigate whether selecting only a particular dependency relation—i.e. using as the input both parent and child words from that dependency relation—leads to clusters with higher semantic relatedness. Each relation can be characterized as either a first- or a second-order relation.[13] A second-order relation is between two words with an intervening preposition, e.g. between a verb and a noun of a directional complement introduced by a preposition, such as in the Dutch "eten achter pc" ("eating at the computer").[14] We ran clustering for each of the forty-five dependency relations separately and measured the quality of each resulting clustering. The cumulative baseline that does not distinguish between dependency relations is given as ALL for first-order relations in Table 4. This is the same result as reported on the first line in Table 2. The addition of second-order dependencies does not change the clustering quality of the baseline (0.196) but increases the number of types.

In the upper part of Table 4, we list six relations leading to clustering quality above the baseline.

[13]The experiments in previous sections included only first-order relations.

[14]The preposition should be seen only as an implicit link between two words and is not included in the input data for clustering. For the example fragment only "eating" and "computer" constitute the data instance actually used by the algorithm.

Type	Ord-1	Ord-2	DepBrown	Population
OBJ2		■	0.238	1,622
LD	■		0.233	2,419
PC		■	0.211	21,157
LD		■	0.208	12,149
OBJ1	■		0.203	108,037
SU	■		0.199	79,844
ALL	■		0.196	495,479
ALL	■	■	0.196	559,908
SU+OBJ1	■		0.202	156,645

Table 4: Lin similarity scores for dependency Brown clustering (*DepBrown*) per type of dependency relation. Ord-1: first-order relation; Ord-2: second-order relation (with intervening preposition); Population: number of word types in the clustering.

Two conclusions can be drawn from the results on these relations. First, some dependency relations contribute better context that leads to increased semantic relatedness compared to clustering without relation selection. Second, both first- and second-order relations appear among the relations outperforming the baseline. The highest score from the top six relations is achieved by taking words exclusively from the second-order secondary object (OBJ2) relation. However, relatively few word types are included in the clusters. The same is true for the first-order directional complements (LD). Of course, clustering with only one of these relations would have quite limited applicability if used in a supervised NLP task due to the low number of word types. However, the main point we want to make here is that these relations yield semantically superior clusters and demonstrate that syntactic functions truly merit further attention in learning semantic clusters using syntax. The remaining four among the top six relations are more frequent relations, and lead to clusterings with higher number of word types. These are the second-order prepositional complement (PC) and directional complement (LD) relations, and the first-order direct object (OBJ1) and subject (SU) relations. Finally, the setting SU+OBJ1 joins words obtained from subject and direct object relations, and achieves a quality that falls between the values obtained for the two relations separately, yet still increases the number of word types.

6 Conclusion and future work

We have presented a detailed study on a simple extension of Brown clustering with a dependency language model. In the first part, we have consolidated the advantage of dependency clustering over standard Brown clustering in a series of experiments, including cluster capacity, granularity level, frequency thresholding, amount of data and other. In the second part, we put forward the idea of selective clustering using data obtained only from specific dependency relations. Several relations lead to a clustering with improved intracluster similarity. We make the code as well as the induced clusters freely available at `https://github.com/rug-compling/dep-brown-cluster`.

Our findings from the selective clustering warrant the development of more complex models capable of including syntactic functions for obtaining semantic clusters. We reserve this work for the future. We find it interesting to apply dependency Brown clustering to languages of different families and compare it in this setting to the standard Brown clustering. The future work further includes a study of the effect of dependency clusters in downstream tasks. Another important point is the effect of parser accuracy on the quality of obtained clusters.

Acknowledgments

Thanks to Çağrı Çöltekin, Gregory Mills, Olga Yeroshina and the anonymous reviewers for valuable suggestions, and to Percy Liang for implementation-related comments.

References

Jordan Boyd-Graber and David M. Blei. 2008. Syntactic topic models. In *NIPS*.

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Marie Candito and Benoît Crabbé. 2009. Improving generative statistical parsing with semi-supervised word clustering. In *IWPT*.

Eugene Charniak. 2001. Immediate-head parsing for language models. In *ACL*.

Wenliang Chen, Min Zhang, and Haizhou Li. 2012. Utilizing dependency language models for graph-based dependency parsing models. In *ACL*.

Grzegorz Chrupala. 2011. Efficient induction of probabilistic word classes with LDA. In *IJCNLP*.

Manaal Faruqui and Chris Dyer. 2013. An information theoretic approach to bilingual word clustering. In *ACL*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *WWW*.

Edouard Grave, Guillaume Obozinski, and Francis Bach. 2013. Hidden Markov tree models for semantic class induction. In *CoNLL*.

Gholamreza Haffari, Marzieh Razavi, and Anoop Sarkar. 2011. An ensemble model that combines syntactic and semantic clustering for discriminative dependency parsing. In *ACL*.

Fei Huang, Alexander Yates, Arun Ahuja, and Doug Downey. 2011. Language models as representations for weakly-supervised nlp tasks. In *CoNLL*.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple Semi-supervised Dependency Parsing. In *Proceedings of ACL-08: HLT*, pages 595–603, Columbus, Ohio.

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

Dekang Lin. 1998. An information-theoretic definition of similarity. In *ICML*.

Scott Miller, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In *HLT-NAACL*.

Saeedeh Momtazi, Sanjeev Khudanpur, and Dietrich Klakow. 2010. A comparative study of word co-occurrence for term clustering in language model-based sentence retrieval. In *ACL-HLT*.

Anjan Nepal and Alexander Yates. 2014. Factorial Hidden Markov models for learning representations of natural language. In *ICLR*.

Gertjan Van Noord. 2006. At Last Parsing Is Now Operational. In *TALN*.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *HLT-NAACL*.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33:161–199.

Barbara Plank and Alessandro Moschitti. 2013. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *ACL*.

Barbara Plank and Gertjan van Noord. 2010. Grammar-driven versus data-driven: Which parsing system is more affected by domain shifts? In *NLPLING Workshop*.

Kashyap Popat, Balamurali A.R, Pushpak Bhattacharyya, and Gholamreza Haffari. 2013. The haves and the have-nots: Leveraging unlabelled corpora for sentiment analysis. In *ACL*.

Martin Popel and David Mareček. 2010. Perplexity of n-gram and dependency language models. In *TSD*.

Kenji Sagae and Andrew S. Gordon. 2009. Clustering words by syntactic similarity improves dependency parsing of predicate-argument structures. In *IWPT*.

David Sánchez, Montserrat Batet, and David Isern. 2011. Ontology-based information content computation. *Knowledge-Based Systems*, 24(2):297–303.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *ACL*.

Yongzhe Shi, Wei-Qiang Zhang, Jia Liu, and Michael Johnson. 2013. Rnn language model with word clustering and class-based output layer. *EURASIP Journal on Audio, Speech, and Music Processing*, (1).

Andreas Stolcke. 2002. SRILM—an extensible language modeling toolkit. In *ICSLP*.

Ang Sun, Ralph Grishman, and Satoshi Sekine. 2011. Semi-supervised relation extraction with large-scale word clustering. In *HLT-ACL*.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *HLT-NAACL*.

Ivan Titov and Alexandre Klementiev. 2012. A Bayesian approach to unsupervised semantic role induction. In *EACL*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *ACL*.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Tim van de Cruys. 2010. *Mining for Meaning: The Extraction of Lexico-semantic Knowledge from Text*. Ph.D. thesis, University of Groningen.

Piek Vossen, Isa Maks, Roxanne Segers, Hennie van der Vliet, Marie-Francine Moens, Katja Hofmann, Erik Tjong Kim Sang, and Maarten de Rijke, editors, 2013. *Cornetto: A Combinatorial Lexical Semantic Database for Dutch*. Springer.