Efficient induction of probabilistic word classes with LDA

Grzegorz Chrupała
Spoken Language Systems
Saarland University
gchrupala@lsv.uni-saarland.de

Abstract

Word classes automatically induced from distributional evidence have proved useful many NLP tasks including Named Entity Recognition, parsing and sentence retrieval. The Brown hard clustering algorithm is commonly used in this scenario. Here we propose to use Latent Dirichlet Allocation in order to induce soft, probabilistic word classes. We compare our approach against Brown in terms of efficiency. We also compare the usefulness of the induced Brown and LDA word classes for the semi-supervised learning of three NLP tasks: fine-grained Named Entity Recognition, Morphological Analysis and semantic Relation Classification. We show that using LDA for word class induction scales better with the number of classes than the Brown algorithm and the resulting classes outperform Brown on the three tasks.

1 Introduction

Word classes automatically induced from distributional evidence have proved useful in a variety of tasks, including Named Entity Recognition (Miller et al. 2004, Ratinov and Roth 2009, Chrupała and Klakow 2010, Turian et al. 2010), parsing (Koo et al. 2008, Suzuki et al. 2009, Candito and Crabbé 2009) and sentence retrieval (Momtazi and Klakow 2009).

Brown et al. (1992) introduced an algorithm which assigns word types to disjoint clusters and it remains a common choice when a simple way to automatically obtain word classes is needed. We present a word class induction method using Latent Dirichlet Allocation (Blei et al. 2003) which has attractive properties compared to Brown:

- It induces a soft, probabilistic clustering on both word types and context features.
- It runs in time linear in the number of classes.

The model maps straightforwardly to the standard document topic model, and thus has the advantage of many existing high quality implementations. We evaluate the model's usefulness on fine-grained Named Entity Recognition (NER), Morphological Analysis (MA) and semantic Relation Classification (RC) and show that

- while the word classes obtained perform better than Brown classes,
- they can be induced in a fraction of the time necessary to run the equivalent Brown model.

2 Inducing word representations

There is a variety of approaches to inducing word representations from distributional information. In this section we briefly review the research most relevant to our proposed approach.

Hard classes Brown et al. (1992) introduced an early model which induces a mapping from word types to classes. It is an agglomerative clustering algorithm which starts with K classes for the K most frequent word types and then proceeds by alternately adding the next most frequent word to the class set and merging the two classes which result in the least decrease in the mutual information between class bigrams. The result is a class hierarchy with word types at the leaves. The overall runtime of the algorithm is $O(K^2V)$ where K is the number of classes and V the number of word types. Lin and Wu (2009) use a distributed version of K-Means to assign words and phrases to hard classes, and successfully use them as features in a NER task and in query classification.

Soft classes A limitation of the Brown model is that it performs hard clustering of word types, and cannot be used to disambiguate word occurrences based on context. Hidden Markov Models have been used to induce probabilistic (soft) word classes: training an HMM on unlabeled data one obtains classes which correspond to multinomial distributions over the vocabulary (Goldwater and Griffiths 2007, Gao and Johnson 2008). Griffiths et al. (2005) propose a model factored into an HMM which generates function words and an LDA topic models which generates content words. Learning the parameters of a bigram HMM takes $O(K^2N)$ time where N is the number of word tokens in the corpus.

Other approaches Other approaches to inducing word representations do not rely on the notion of word class. Distributed word embeddings can be learned using a neural network-bases language models (Bengio et al. 2006, Collobert and Weston 2008, Mnih and Hinton 2009). Dimensionality reduction techniques such as SVD (Schütze 1995, Lamar et al. 2010) and LSA (Deerwester et al. 1990) have also been found useful for generating word representations.

3 Using word representations

Our main motivation for studying word class induction methods is to use them in a semi-supervised learning scenario, where word representations are induced from a large unlabeled corpus and subsequently used as a source of features for a supervised model. Turian et al. (2010) compare the effect of using representations based on Brown classes, the Collobert and Weston (2008) embeddings and the Mnih and Hinton (2009) embeddings in learning English syntactic chunking (CoNLL 2000) and English coarse-grained Named Entity Recognition (CoNLL 2003). For both tasks the best representation is fine-grained Brown classes (3200 and 1000 classes respectively). Combining the Brown features with distributed embeddings further improves performance on NER but not on chunking. Lin and Wu (2009) use induce word and phrase classes and report results on NER which are higher than Turian et al. (2010)'s Brown scores, but this research used 700 billion words of web text and needed a cloud computing infrastructure with 1000 CPUs to run. It is evident that the Brown clustering algorithm still provides an extremely competitive baseline nearly 20 years after it was proposed.

We thus compare the performance of the LDA word class model to the Brown model on three NLP tasks: fine grained Named Entity Recognition, Morphological Analysis, and Relation Classification. The first two tasks are difficult due to the large number of labels and high potential for ambiguity. The third task is challenging for a different reason: it involves highly abstract semantic relations, often not obviously inferable from surface lexical clues.

4 LDA model for word class induction

We propose an LDA-based model for word class induction and contrast its structure, efficiency, and performance to those exhibited by the Brown model.

4.1 Weaknesses of Brown

Here we address what we see as two related weaknesses of the Brown model. The algorithm's quadratic dependence on K makes it inconvenient to induce more than a few hundred classes: running a 1.000 class model with a 400.000 vocabulary took over 100 hours. Second, the induced clustering is hard, and the only way to model ambiguous word types is to have a separate class for each kind of ambiguity. This in turn means that we need to learn a large number of classes, which exacerbates the problem with inefficiency. Very fine-grained Brown classes are typically needed for good performance as shown by Turian et al.'s results. Our model for word class induction addresses both of the weaknesses.

4.2 LDA for word class induction

Latent Dirichlet Allocation (LDA) was initially introduced by Blei et al. (2003) in the context of topic modeling, i.e. finding coherent topics shared among subsets of a collection of documents. LDA is a generative, probabilistic hierarchical Bayesian model which induces a set of latent variables which correspond to the topics. The topics themselves are multinomial distributions over words. The graphical model is shown in plate notation in Figure 1. The generative structure of the LDA model is as follows:

$$\begin{aligned}
\phi_k &\sim \text{Dirichlet}(\beta), & k &\in [1, K] \\
\theta_d &\sim \text{Dirichlet}(\alpha), & d &\in [1, D] \\
z_{n_d} &\sim \text{Multinomial}(\theta_d), & n_d &\in [1, N_d] \\
w_{n_d} &\sim \text{Multinomial}(\phi_{z_{n_d}}), & n_d &\in [1, N_d]
\end{aligned} \tag{1}$$

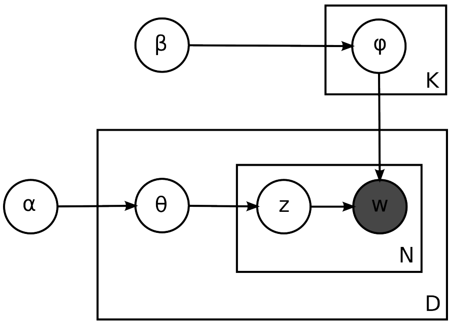


Figure 1: LDA plate diagram

The document collection is generated by drawing, for each topic k, a distribution over words ϕ_k from a Dirichlet prior with parameters β. Then for each document d we draw a multinomial distribution over topics θ_d from a Dirichlet prior parametrized by α. To generate the n^{th} word in document d we draw the topic id z_{n_d} from the document-specific topic distribution θ_d, and then draw the word from the word distribution corresponding to the chosen topic $\phi_{z_{n_d}}$. Thus each document is a mixture of different topics, giving the model the flexibility needed to reflect the topical structures in real-world document collections. This flexibility has contributed to the popularity of LDA as a common choice in a wide range of domains beyond topic modeling. Another important reason for LDA's success is the availability of efficient and well-understood estimation methods such as Variational EM (Blei et al. 2003), and Gibbs sampling (Griffiths and Steyvers 2004). For both methods efficient, well engineered and well tested implementations are readily available. These advantages have led us to try to use a model equivalent to an LDA topic model in order to induce word classes based on distributional clues.

We associate each word type with a distribution over latent classes. Each class is in turn a distribution over contextually co-occurring features. In principle the contextual features could be arbitrary functions of the context, but to make our model use exactly the same information as the Brown model, we will restrict them to the word's immediate left and right neighbors. A direct mapping to document topic model can be seen:

Topic model	Word class induction
Document	Word type
Word	Context feature
Topic	Word class

An example "document" in our scenario, corresponding to the word type *Krzysztof* looks like the following:

Bledkowski$_R$ **Kieslowski**$_R$ **Kieslowski**$_R$ **Rutkowski**$_R$ **Sikorski**$_R$ **and**$_L$ **argues**$_L$ **argues**$_R$ **director**$_L$ **director**$_L$ **edits**$_R$ **said**$_R$

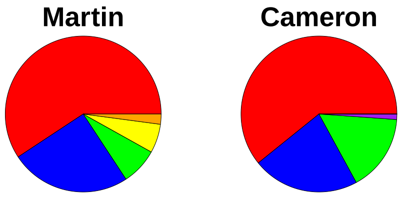
The subscript on the word indicates whether it is a left or right context feature, i.e. whether it appears to left or to the right of *Krzysztof* in the corpus.

Thus, strictly speaking our model does not generate the actual sequence of words in the corpus, but rather a collection of "documents" such as the above, or equivalently, a table listing bigram co-occurrence counts for each word type.

The generative structure of the model corresponds exactly to a standard LDA topic model in equation 1. Now K is the number of latent classes, D is the vocabulary size, and N_d is the number of left and right contexts in which word type d appears, z_{n_d} is the class of word type d in the n_d^{th} context, and f_{n_d} is the n_d^{th} context feature of word type d.

Once trained, the parameters provide two types of word representations. Each θ_d gives the latent class probability distribution given a word type. Each ϕ_k gives the feature distribution given a latent class. Thus the model provides a probabilistic representation for word types independently of their context, and also for contexts independently of the word type. This is a more powerful representation than hard word clustering: (i) soft clustering allows the modeling of ambiguity, (ii) additional source of information is available which helps determine the class of a word from its context.

Figure 2 shows an example of how the classes discovered by the model deal with ambiguity. The pie charts depict the induced class distributions for two word types *Martin* and *Cameron*. These words are ambiguous in a similar way: they are mostly used as (i) first names or (ii) family names, and (iii) additionally can appear as part of a name of a company or place. This similarity of usage is reflected in closeness of the distributions over the induced classes for those words. Thus the first class (colored red) is associated with many family names, the second (blue) with titles and first names, and the third (green) with companies and locations. This correspondence is certainly not



chief Gingrich Martin Newt Van Scott Roberts
Mr. Ms. John Robert President Dr. David
Street General Texas Fidelity State California

Figure 2: Class distributions for the word types *Martin* and *Cameron*. Also shown are the most common word types for the three largest.

Newt, Speaker	●	executive, operating
says, Chairman	●	Clinton, Dole, J.
Wall, West, East	●	County, AG, Journal

Figure 3: Most frequent left and right context-word features co-occurring with the three classes from Figure 2

perfect (e.g. the first class is also associated with the word *chief*) but it is suggestive that soft LDA-based clustering can successfully model and discover this type of systematic shared ambiguities.

We can use the same three classes to illustrate the second advantage of LDA word classes mentioned above: we can obtain information about the class of a particular token based on its context. Thus even for a rare word which did not appear in the corpus used for word class induction, we can still find out what word classes it is associated with just by consulting the ϕ table and retrieving the classes strongly associated with the context features. Figure 3 shows the left and right context features which co-occurred most frequently with the same three classes illustrated in Figure 2. For example the second (blue) class, which contains titles and first names, is associated with left contexts such as *says* and *Chairman* and right contexts such as *Clinton* and *Dole*.

5 Experimental evaluation

In order to evaluate the LDA word class induction model we assess two of its aspects: (i) we compare its efficiency to that of Brown clustering, and (ii) we compare the performance of the induced word classes to those obtained by Brown clustering in two difficult sequence labeling tasks and one classification task.

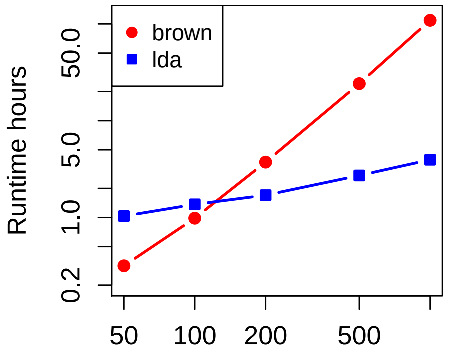


Figure 4: Brown and LDA run times

5.1 Efficiency

Two main approaches have been used to train LDA topic models: variational EM (Blei et al. 2003) and Gibbs sampling (Griffiths and Steyvers 2004). Both scale linearly with the number of topics. This property is one of the main advantages of the LDA word class induction model over HMM or Brown clustering. In this section we show that also in practice this means that LDA word classes can be induced much faster than equivalently performing Brown classes.

As training data for both models we use the North American News Text Corpus (over 60M words). For both models we only keep bigrams which occur at least 3 times. The resulting vocabulary is over 380K word types. The LDA class induction model scales as $O(KN)$ where N is the sum of all feature counts. Since we discard rare bigrams with frequencies under m, we can scale the remaining feature counts by $1/m$ and obtain an equivalent model, while reducing runtime by m times. For both models we induce 50, 100, 200, 500 and 1000 classes. For Brown we run the implementation of Liang (2005) until termination. For LDA we run 1000 iterations of a collapsed Gibbs sampler from Mallet (McCallum 2002). Figure 4 shows how the models scale with growing value of K on a log-log plot. Brown terminates in 20 minutes for K=50, but takes over 110 hours for K=1000, while LDA takes between 1 hour for K=50 and 4 hours for K=1000.

5.2 Performance

Another advantage of LDA word classes over hard clusters is the increased representational power. Ambiguity can be modeled more compactly, and there are two sources of information to draw on when deciding on the most likely class of a word in context: the word type identity, and the local context features. In this section we show that these theoretical advantages translate into performance on fine-grained Named Entity Recognition, Morphological Analysis and on semantic Relation Classification.

In each of the tasks we tried to use the Brown classes and the LDA classes in an optimal way by taking advantage of the strength of each type of representation, and also to adapt the feature sets to the specifics of the task. We followed previous work when available and ran exploratory experiments with different feature combinations on the development data. The details of the final feature sets are given in the respective sections but in general we make use of the hierarchical nature of Brown classes by using them at several levels of granularity. For LDA classes we exploit their probabilistic softness by including feature probability or rank, and include classes inferred from context words when appropriate.

5.2.1 Named entity recognition

Named-entity recognition is one of the most commonly needed NLP task. Many evaluations have focused on learning the coarse-grained CoNLL and MUC entity labels (person, organization and location). Here we evaluate on the more challenging fine-grained entities from the BBN corpus (Weischedel and Brunstein 2005). We use sections 2 to 21 as training data, section 22 for development and section 23 for final evaluation. We keep all labels appearing at least 100 times in training data. Less frequent labels we map to an existing more generic label if possible (e.g. LOCATION:LAKE_SEA_OCEAN ↦ LOCATION:OTHER), otherwise we discard them. We also discard all **description** labels which are not proper named entities. We are left with 40 labels, shown in Table 1.

We convert the labeling to the BIO format which encodes chunking information into token-level labels: each label is prefixed with B if it starts a new chunk, I if it continues the previous chunk, or O if it does not belong to a named entity chunk. Thus we end up with 81 labels after conversion.

lowercase	Map all characters to lower case
wordshape	Encodes spelling of a token by mapping sequences of upper case letters to `X`, lower case letters to `x`, digits to `0`, hyphens and underlines to themselves. For example *IJCNLP-2011* maps to `X-0`
suffix$_n$	The n characters from the end of the token
rank$^n_z f(z)$	The n^{th} class in the ranking ordered by the value of the function f
prefix$_n$	The first n characters from the start
z	Class id

Table 2: Meaning of feature functions

Baseline As a baseline we use a sequence-perceptron labeler (Collins 2002) with the following features: $\{w_{-2}, w_{-1}, w_0, \text{lowercase}(w_0), \text{wordshape}(w_0), \text{suffix}_1(w_0), \text{suffix}_2(w_0), \text{suffix}_3(w_0), w_1, w_2\}$. For the explanation of the feature functions see Table 2.

For inducing classes for this task we use the North American News Text Corpus described in section 5.1. When evaluating word classes we add to this feature set the Brown or LDA word class features:

Brown Class IDs encode the path in the class hierarchy, we thus use ID prefixes of different lengths to include classes at several levels of granularity. We also add feature conjunctions. The additional Brown class features are thus: $\text{prefix}_n(z(w_m))$ (for tokens at positions $m \in \{-1, 0, 1\}$, class id prefix of length n for $n \in \{4, 6, 10, 20\}$) and feature conjunctions $\{\text{prefix}_{20}(z(w_0))\} \times \{\text{lowercase}(w_0), \text{wordshape}(w_0), \text{suffix}_1(w_0), \text{suffix}_2(w_0), \text{suffix}_3(w_0)\}$. The class ID prefix sizes we adopted were shown to be effective in Ratinov and Roth (2009) and Turian et al. (2010).

LDA We rank the classes according to posterior probability and take the 3 top ranked classes given the current word d, the 1 top ranked class given the previous word w_{-1}, and 1 top ranked class given the next word w_{+1}: $\{\text{rank}^1_z P(z|d), \text{rank}^2_z P(z|d), \text{rank}^3_z P(z|d), \text{rank}^1_z P(z|w_{-1}), \text{rank}^1_z P(z|w_{+1})\}$. We add the following feature conjunctions: $\{\text{lowercase}(w_0), \text{wordshape}(w_0), \text{suffix}_1(w_0), \text{suffix}_2(w_0), \text{suffix}_3(w_0)\} \times \{\text{rank}^1_z P(z|d),$

ANIMAL CARDINAL DATE:AGE DATE:DATE DATE:DURATION DATE:OTHER DISEASE EVENT:OTHER FAC:BUILDING
FAC:HIGHWAY-STREET GPE:CITY GPE:COUNTRY GPE:OTHER GPE:STATE-PROVINCE LAW LOCATION:CONTINENT
LOCATION:OTHER LOCATION:REGION MONEY NORP:NATIONALITY NORP:POLITICAL ORDINAL
ORGANIZATION:CORPORATION ORGANIZATION:EDUCATIONAL ORGANIZATION:GOVERNMENT
ORGANIZATION:OTHER ORGANIZATION:POLITICAL PERCENT PERSON PLANT PRODUCT:OTHER PRODUCT:VEHICLE
QUANTITY:1D QUANTITY:WEIGHT SUBSTANCE:CHEMICAL SUBSTANCE:DRUG SUBSTANCE:FOOD
SUBSTANCE:OTHER TIME WORK-OF-ART:OTHER

Table 1: BBN named entity labels

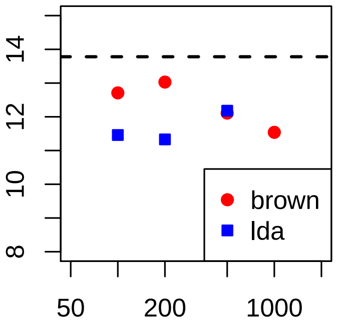


Figure 5: F1 error on NER dev. set with word classes

rank$^1_z P(z|w_{-1})$, rank$^1_z P(z|w_{+1})\}$.

Figure 5 shows the F1 error on section 22 for the baseline and for Brown and LDA classes of different granularity. A large number of classes (500 or 1000) is needed to achieve low error with Brown classes. With LDA, a lower number (100 or 200) is sufficient, and in fact the error rates are lower for LDA word class features than for Brown. The results for the test set (section 23) using Brown with 1000 classes and LDA with 200 classes are shown in the *NER* column of Table 5. We are unaware of previous published results on BBN at a comparable level of NE label granularity.

5.2.2 Morphological analysis

We next evaluate the induced word classes on a morphological analysis task. The goal is to learn to assign morpho-syntactic descriptors (MSD) (roughly speaking, fine-grained POS tags) and lemmas to tokens in sentences. The MSD tags encode all relevant inflectional features of a token such as gender, case and number for nouns or tense, aspect, person and number for verbs. A morphological tagger which performs this type of analysis is an important component for processing languages with rich inflectional morphology. Figure 6 shows example morphological annotation of a short sentence in Spanish.

Token	Lemma	MSD	Gloss
Pero	pero	`cc`	but
cuando	cuando	`cs`	when
era	ser	`vsii3s0`	he was
niño	niño	`ncms000`	boy
le	el	`pp3csd00`	to him
gustaba	gustar	`vmii3p0`	it pleased

Figure 6: Morphosyntactic annotation of a Spanish which translates as *When he was a boy he liked it.*

As the supervised model we use the Morfette system (Chrupała et al. 2008)[1]. Morfette trains two classifiers, one for morphological tags (i.e. fine-grained POS tags) and one for lemmatization classes. The classifiers are trained separately; their output is combined during decoding. For the baseline we used the default features (see Chrupała et al. 2008) and trained the POS and lemma models for 10 and 3 iterations respectively. We added word-class features to the POS model.

Brown We use class id prefixes for the focus word: prefix$_n(z(w_0))$, $n \in \{4, 6, 10, 20\}$

LDA Morfette can use real-valued features and initial tests on this task with class distributions showed that using them directly works as well as discretizing them. We use classes for the focus token (w_0) and set probabilities below 0.15 to 0 for sparseness.

Data sets We chose two data sets for evaluation:

- Spanish Ancora corpus (Taulé et al. 2008): portion corresponding to data used by Chrupała et al. (2008), 188.803 tokens, 10.000 dev. and 10.000 test, 279 tags.

[1] Available at `http://code.google.com/p/morfette/`

CAUSE-EFFECT INSTRUMENT-AGENCY
PRODUCT-PRODUCER CONTENT-CONTAINER
ENTITY-ORIGIN ENTITY-DESTINATION
COMPONENT-WHOLE MEMBER-COLLECTION
COMMUNICATION-TOPIC

Table 3: Relation classification labels

- French Treebank (Abeillé et al. 2003), 351.873 tokens, 36.297 dev. and 37.967 test, 214 tags.

For inducing word classes we used (i) Spanish Europarl (Koehn 2005), 50M words and (ii) Est Republicain[2] 147M words.

We optimized the number of classes on the development set: for Brown the best was 500 for both languages, for LDA the best setting was 50 for Spanish and 100 for French.

Table 5 (columns *MA es* and *MA fr*) shows the joint morphological tagging-lemmatization scores on the test set. Word classes give a moderate performance boost and in both cases LDA improves more. We do not know of published results on the French data with this level of granularity. However Seddah et al. (2010) show that Morfette on French data with a reduced tagset compares well to state-of-the-art, and thus can be assumed to be a strong baseline. For Spanish, our baseline error is almost identical to the error reported by Chrupała et al. (2008) (4.98 vs 5.00): thus the word classes give 10% relative error reduction over previous results.

5.2.3 Classification of semantic relations

The last task on which we evaluate induced word classes is multi-way classification of abstract semantic relations between nominals (RC). This task appeared at the Semeval 2007 and 2010 workshops. We use the task definition and the training and testing data from 2010.

The relation inventory is shown in Table 3. For example in the sentence *The bowl was full of apples, pears and oranges* the nominal *pears* is in a CONTENT-CONTAINER relation with the nominal *bowl*.

The training set consists of sentences annotated with the relations and their directionality. The arguments (nominals) participating in the relations are marked in both the training and test examples. We used the 2010 training set of 8000 sentences and the test set of 2717 sentences. During development, we split the training set in half, and trained on the first half, while validating on the second half. For the final evaluation we trained on all the 8000 training sentences.

arg_1	The first argument
arg_2	The second argument
between_n	Each of the tokens between arg_1 and arg_2
before_m	Each of the 3 tokens before arg_1
after_m	Each of the 3 tokens after arg_2

Table 4: Description of features for RC

We evaluated with the scoring script provided, using the official macro-averaged F1 score.

Baseline For our baseline system we used the Weka (Hall et al. 2009) implementation of the Sequential Minimal Optimization algorithm to train an SVM classifier (Platt 1999), with the default linear kernel. We treat each combination of the relation label and the direction label together as a single atomic class to be learned.

Table 4 describes the features we extracted from each sentence for the RC task.

Corpus For this task we initially used the word classes induced from The North American News Text Corpus described in section 5.1. The improvements we achieved were smaller than we expected. We suspected that the reason for this was that the training data for the RC task come from a variety of Web sources and are much less restricted in genre than the text in the NANT corpus. We thus decided to retrain word classes on the more balanced 100M-word BNC corpus (BNC Consortium 2001). As expected, the word classes from the BNC worked better. Due to time constraints, for Brown we were able to induce only up to 500 classes.

Brown With Brown classes we use class ID prefix $\text{prefix}_n(z(f))$, $n \in \{4, 6, 10, 20\}$, for each of the baseline features f listed in Table 4.

LDA For this task we use the probabilities of the classes as real-valued features, and we took classes for each of the baseline features f listed Table 4.

We optimized the number of classes on the development data (second half of training data). We found 500 classes for Brown and 100 classes for

[2] http://www.cnrtl.fr/corpus/estrepublicain/

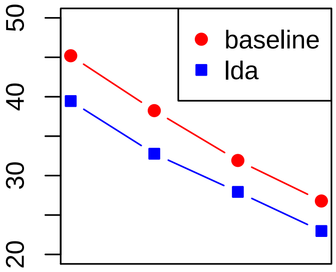


Figure 7: Relation classification error as a function of the number of training examples. The x-axis is plotted on a logarithmic scale.

LDA to give the best results, and we used these values for the final evaluation.

The impact of adding word classes can be appreciated in Figure 7, which plots the test error with and without word classes while varying the number of training examples (1000, 2000, 4000 and 8000). It can be seen that adding LDA word class features corresponds to almost doubling the amount of training data.

Table 5 (column labeled *RC*) shows the macro-averaged F1 error on test data. Similarly to the previous tasks, the improvement is larger with LDA than with Brown classes.

For comparison, during the Semeval 2010 evaluation the F1 error of the top-scoring system (Rink and Harabagiu 2010) was 17.81%; the system ranked second (Tymoshenko and Giuliano 2010) achieved 22.37%.

Both these systems used large amounts of external resources and heavy-duty linguistic processing tools. Rink and Harabagiu (2010) extracted features from dependency parses, from PropBank and FrameNet parses, from WordNet and NomLex as well as using Google n-grams and the output of TextRunner[3]. Tymoshenko and Giuliano (2010) extracted features from syntactic parses and from the massive semantic knowledge database Cyc (Lenat 1995).

In comparison, our system is extremely resource-light since our features do not rely on any manually created databases or linguistic processing tools (not even POS tags). It is thus satisfying that by automatically and efficiently inducing simple word class features we can achieve results which are close to state-of-the-art.

	%Error			
Model	NER	MA es	MA fr	RC
Baseline	13.42	5.00	7.80	26.78
Brown	11.82	4.70	7.51	25.66
LDA	**11.70**	**4.50**	**7.39**	**22.97**

Table 5: Test set results on NER, MA, RC

6 Discussion

To our knowledge LDA word class induction has not been previously used in this particular scenario. LDA variants have been proposed in other settings: Brody and Lapata (2009) use LDA to induce latent variables corresponding to word senses; Toutanova and Johnson (2007) propose an LDA-inspired model where induced word-classes are used for semi-supervised POS tagging; Dinu and Lapata (2010) use LDA-induced word-classes for measuring word similarity in context. Rather than focus on adapting LDA to a particular task, we instead induce generic word classes that can be plugged in as features in a number of NLP applications.

We show that the LDA word clustering algorithm is an attractive choice for semi-supervised learning. It is efficient to train and beats a competitive baseline provided by Brown clustering.

Acknowledgements

This work was carried out in the context of the Software-Cluster project EMERGENT (www.software-cluster.org). It was partially funded by the German Federal Ministry of Education and Research (BMBF) under grant number 01IC10S01O. The author assumes responsibility for the content.

References

Abeillé, A., Clément, L., and Toussenel, F. (2003). Building a treebank for French.

Bengio, Y., Schwenk, H., Senécal, J., Morin, F., and Gauvain, J. (2006). Neural Probabilistic Language Models. *Innovations in Machine Learning*, pages 137–186.

Blei, D., Ng, A., and Jordan, M. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

[3] A system for open information extraction from the Web (Yates et al. 2007).

BNC Consortium (2001). The British National Corpus, version 2 (BNC World). Distributed by Oxford University Computing Services on behalf of the BNC Consortium. http://www.natcorp.ox.ac.uk/.

Brody, S. and Lapata, M. (2009). Bayesian word sense induction. In *EACL 2009*.

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

Candito, M. and Crabbé, B. (2009). Improving generative statistical parsing with semi-supervised word clustering. In *IWPT 2009*.

Chrupała, G., Dinu, G., and Van Genabith, J. (2008). Learning morphology with Morfette. In *LREC 2008*.

Chrupała, G. and Klakow, D. (2010). A Named Entity Labeler for German: exploiting Wikipedia and distributional clusters. In *LREC 2010*.

Collins, M. (2002). Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *ACL 2002*.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML 2008*.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Dinu, G. and Lapata, M. (2010). Measuring distributional similarity in context. In *EMNLP 2010*.

Gao, J. and Johnson, M. (2008). A comparison of Bayesian estimators for unsupervised Hidden Markov Model POS taggers. In *EMNLP 2008*.

Goldwater, S. and Griffiths, T. (2007). A fully Bayesian approach to unsupervised part-of-speech tagging. In *ACL 2007*.

Griffiths, T. and Steyvers, M. (2004). Finding scientific topics. *PNAS*, 101(Suppl 1):5228.

Griffiths, T., Steyvers, M., Blei, D., and Tenenbaum, J. (2005). Integrating topics and syntax. In *NIPS 2005*.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. (2009). The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT Summit 2005*.

Koo, T., Carreras, X., and Collins, M. (2008). Simple semi-supervised dependency parsing. In *ACL 2008*.

Lamar, M., Maron, Y., Johnson, M., and Bienenstock, E. (2010). SVD and clustering for unsupervised POS tagging. In *ACL 2010*.

Lenat, D. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.

Liang, P. (2005). *Semi-supervised learning for natural language*. PhD thesis, Massachusetts Institute of Technology.

Lin, D. and Wu, X. (2009). Phrase clustering for discriminative learning. In *ACL/IJCNLP 2009*.

McCallum, A. (2002). Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Miller, S., Guinness, J., and Zamanian, A. (2004). Name tagging with word clusters and discriminative training. In *HLT/NAACL 2004*.

Mnih, A. and Hinton, G. (2009). A scalable hierarchical distributed language model. In *NIPS 2009*.

Momtazi, S. and Klakow, D. (2009). A word clustering approach for language model-based sentence retrieval in Question Answering systems. In *CIKM 2009*.

Platt, J. (1999). Sequential minimal optimization: A fast algorithm for training support vector machines. *Advances in Kernel Methods-Support Vector Learning*, 208:98–112.

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *CoNLL 2009*.

Rink, B. and Harabagiu, S. (2010). Utd: Classifying semantic relations by combining lexical and semantic resources. In *SemEval 2010*, pages 256–259.

Schütze, H. (1995). Distributional part-of-speech tagging. In *ACL 1995*.

Seddah, D., Chrupała, G., Çetinoglu, Ö., van Genabith, J., and Candito, M. (2010). Lemmatization and Lexicalized Statistical Parsing of Morphologically Rich Languages: the Case of French. In *SPMRL, NAACL workshop*.

Suzuki, J., Isozaki, H., Carreras, X., and Collins, M. (2009). An empirical study of semi-supervised structured conditional models for dependency parsing. In *EMNLP 2009*.

Taulé, M., Martí, M., and Recasens, M. (2008). Ancora: Multilevel annotated corpora for Catalan and Spanish. In *LREC-2008*.

Toutanova, K. and Johnson, M. (2007). A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In *NIPS 2007*.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *ACL 2010*.

Tymoshenko, K. and Giuliano, C. (2010). Fbk-irst: Semantic relation extraction using cyc. In *SemEval 2010*.

Weischedel, R. and Brunstein, A. (2005). BBN pronoun coreference and entity type corpus. Linguistic Data Consortium.

Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., and Soderland, S. (2007). Textrunner: Open information extraction on the web. In *NAACL-HLT 2007 Demonstration Program*.