Unsupervised Induction of Linguistic Categories with Records of Reading, Speaking, and Writing

Maria Barrett[1] **Ana V. González-Garduño**[2]
Lea Frermann[3] **Anders Søgaard**[2]
[1]Department of Nordic Studies and Linguistics, University of Copenhagen, Denmark
[2]Department of Computer Science, University of Copenhagen, Denmark
[3]Amazon Development Center, Berlin, Germany
`barrett@hum.ku.dk {ana, soegaard}@di.ku.dk`
`lfrerman@amazon.com`

Abstract

When learning POS taggers and syntactic chunkers for low-resource languages, different resources may be available, and often all we have is a small tag dictionary, motivating type-constrained unsupervised induction. Even small dictionaries can improve the performance of unsupervised induction algorithms. This paper shows that performance can be further improved by including data that is readily available or can be easily obtained for most languages, i.e., eye-tracking, speech, or keystroke logs (or any combination thereof). We project information from all these data sources into shared spaces, in which the union of words is represented. For English unsupervised POS induction, the additional information, which is not required at test time, leads to an average error reduction on Ontonotes domains of 1.5% over systems augmented with state-of-the-art word embeddings. On Penn Treebank the best model achieves 5.4% error reduction over a word embeddings baseline. We also achieve significant improvements for syntactic chunk induction. Our analysis shows that improvements are even bigger when the available tag dictionaries are smaller.

1 Introduction

It is a core assumption in linguistics that humans have knowledge of grammar and that they use this knowledge to generate and process language. Reading, writing, and talking leave traces of this knowledge and in psycholinguistics this data is used to analyze our grammatical competencies. Psycholinguists are typically interested in falsifying a specific hypothesis about our grammatical competencies and therefore collect data with this hypothesis in mind. In NLP, we typically require big, representative corpora. NLP usually has induced the models from expensive corpus annotations by professional linguists, but recently, a few researchers have shown that data traces from human processing can be used directly to improve NLP models (Klerke et al., 2016; Barrett et al., 2016; Plank, 2016).

In this paper, we investigate whether unsupervised POS induction and unsupervised syntactic chunking can be improved using human text processing traces. We also explore what traces are beneficial, and how they are best combined. Our work supplements psycholinguistic research by evaluating human data on larger scale than usual, but more robust unsupervised POS induction also contributes to NLP for low-resource languages for which professional annotators are hard to find, and where instead, data from native speakers can be used to augment unsupervised learning.

We explore three different modalities of data reflecting human processing plus standard, pretrained distributional word embeddings for comparison, but also because some modalities might fare better when combined with distributional vectors. Data reflecting human processing come from reading (two different eye-tracking corpora), speaking (prosody), and typing (keystroke logging). We test three different methods of combining the different word representations: a) canonical correlation analysis (CCA) (Faruqui and Dyer, 2014b) and b) singular value decomposition and inverted softmax feature projection (SVD+IS) (Smith et al., 2017) and c) simple concatenation of feature vectors.

Contributions We present experiments in unsupervised POS and syntactic chunk induction using multi-modal word representations, obtained from records of reading, speaking, and writing. Individually, all modalities are known to contain syntactic processing signals, but to the best of our

Lea Frermann carried out this work while at the University of Edinburgh.

Proceedings of NAACL-HLT 2018, pages 2028–2038
New Orleans, Louisiana, June 1 - 6, 2018. ©2018 Association for Computational Linguistics

knowledge, we are the first to combine them in one model. Our work extends on previous work in several respects: (a) We compare using data traces from gaze, speech, and keystrokes. (b) We consider three ways of combining such information that do not require access to data from all modalities for all words. (c) While some previous work assumed access to gaze data at test time, our models do not assume access to any modalities at test time. (d) We evaluate how much the additional information helps, depending on the size of the available tag dictionary. (e) While related work on keystrokes and prosody focused on a single feature, all our word representations are multidimensional and continuous.

2 Related work

Eye-tracking data reflect the eye movements during reading and provide millisecond-accurate records of the readers fixations. It is well established that the duration of the fixations reflect the processing load of the reader (Rayner, 1998). Words from closed word classes are usually fixated less often and for shorter time than words from open word classes (Rayner and Duffy, 1988). Psycholinguistics, however, is generally not interested in covering all linguistic categories, and psycholinguists typically do not study corpora, but focus instead on small suites of controlled examples in order to explore human cognition. This is in contrast with NLP. Some studies have, however, tried to bridge between psycholinguistics and NLP. Demberg and Keller (2008) found that eye movements reflected syntactic complexity . Barrett and Søgaard (2015a) and Barrett and Søgaard (2015b) have tried to–respectively–predict a full set of syntactic classes and syntactic functions across domains in supervised setups. Barrett et al. (2016), which is the work most similar to ours, used eye-tracking features from the Dundee Corpus (Kennedy et al., 2003), which has been augmented with POS tags by Barrett et al. (2015). They tried for POS induction both on token-level and type-level features. They found that eye-tracking features significantly improved tagging accuracy and that type-level eye-tracking features helped more than token-level. We use the same architecture as Barrett et al. (2016).

Keystroke logs also reflect the processing durations, but of writing. Pauses, burst and revisions in keystroke logs are used to investigate the cognitive process of writing (Matsuhashi, 1981; Baaijen et al., 2012). Immonen and Mäkisalo (2010) found that for English-Finnish translation and monolingual Finnish text production, predicate phrases are often preceded by short pauses, whereas adpositional phrases are more likely to be preceded by long pauses. Pauses preceding noun phrases grow with the length of the phrase. They suggest that the difference is explained by the processing of the predicate begins before the production of the clause starts, whereas noun phrases and adpositional phrases are processed during writing. Pre-word pauses from keystroke logs have been explored with respect to multi-word expressions (Goodkind and Rosenberg, 2015) and have also been used to aid shallow parsing (Plank, 2016) in a multi-task bi-LSTM setup.

Prosodic features provide knowledge about how words are pronounced (tone, duration, voice etc.). Acoustic cues have already been used to improve unsupervised chunking (Pate and Goldwater, 2011) and parsing (Pate and Goldwater, 2013). Pate and Goldwater (2011) cluster the acoustic signal and use cluster label as a discrete feature whereas Pate and Goldwater (2013) use a quantized word duration feature.

Plank (2016) and Goodkind and Rosenberg (2015) also used a single keystroke feature (keystroke pre-word pause) and the former study also discretized the feature. Our work, in contrast, uses acoustic and keystroke features as multidimensional, continuous word representations.

3 Modalities

In our experiments, we begin with five sets of word representations: prosody, keystroke, gaze as recorded in the GECO corpus, gaze as recorded in the Dundee corpus, as well as standard, text-based word embeddings from eigenwords. See below for details and references. All modalities except the pre-trained word embeddings reflect human processing of language. For all modalities, we use type-level-averaged features of lower-cased word types.

The choice of using type-averaged features is motivated by Barrett et al. (2016), who tried both token-level and type-averaged eye-tracking features for POS induction and found that type-level gaze features worked better than token-level. Type-averaged features also have the advantage of not relying on access to the auxillary data at test

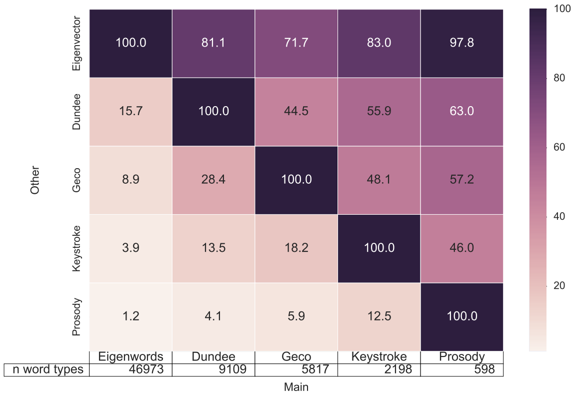


Figure 1: The percentage of overlapping word types for pairs of modalities. Overlapping words are used for projecting word representations into a shared space. Read column-wise. E.g. when combining eigenwords and prosody, only 1.2% of the 46973 eigenvector word types are overlapping (bottom left), and 97.8% of the 598 prosody word types are overlapping (top right).

time. Type-level averages are simply looked up in an embedding file for all previously seen words. On the other hand, type-level features obviously do not represent ambiguities, e.g., *beat* as a verb and a noun separately. All our features, except log-transformed word frequencies were normalized.

We run unsupervised induction experiments for all ($2^5 - 1 = 31$) combinations of our five data sources on the development sets to determine which data types contribute to the task. We consider three different ways of combining modalities, two of which learn a projection into a shared space using word overlap as supervision, and one simply concatenates the embedding spaces. The combination methods are further described in §4.

We list the number of word types per modality and percentage of pair-wise overlapping words in Figure 1. We only use existing data from native speaking participants, for reproducibility and in order not to get learner effects ie. biases introduced by non-native speakers. §3.2-3.5 describe each modality in detail, and how we compute the word representations. §3.1 describes a set of basic features used in all of our experiments.

3.1 Basic features

Like Li et al. (2012), we append a small set of basic features to all our feature sets: features relating to orthography such as capitalization, digits and suffix. Furthermore we append log word frequency and word length. Word frequencies per million are obtained from British National corpus (BNC) frequency lists (Kilgarriff, 1995). Word length and word frequency explain around 70% of the variance in the eye movement (Carpenter and Just, 1983) and are therefore also important for estimating the impact of gaze features beyond such information. Plank (2016) used keystroke features for shallow chunking and did not find any benefit of normalizing word length by pre-word pause before typing each word, but Goodkind and Rosenberg (2015) did find a strong logarithmic relationship between word length and pre-word pause as well as between word frequency and pre-word pause.

Modality	n found pairs	Weigh. av. cor.
Prosody	31	0.369
Keystroke	1082	0.060
GECO	2449	-0.030
Dundee	4066	-0.035
Eigenwords	9828	0.197

Table 1: Results on word association norms from wordvectors.org Correlation weighted by number of found pairs per word embedding type.

3.2 Dundee and GECO eye-tracking corpora

We use two different eye-tracking corpora. The GECO corpus (Cop et al., 2017) and the Dundee Corpus (Kennedy et al., 2003) are the two largest eye movement corpora with respect to word count. We use the native English part of the GECO corpus and the English part of the Dundee Corpus. The GECO corpus is publicly available[1] and the Dundee Corpus is available for research purposes.

Participants and data The Dundee Corpus is described in Kennedy and Pynte (2005). The Dundee Corpus consists of the eye movements of 10 readers as they read the same 20 newspaper articles. For GECO, all 14 participants in the native English part read a full Agatha Christie novel. Both corpora contain $>$ 50.000 words per reader. All participants for both corpora are adult, native speakers of English and skilled readers.

Self-paced reading Both eye-tracking corpora reflect natural reading by making the reading self-paced and using naturally-occurring, contextualized text.

[1] http://expsy.ugent.be/downloads/geco/

Features Eye movements–like most features reflecting human processing–are very susceptible to experiment-specific effects e.g. instructions and order effects such as fatigue. Furthermore, the GECO corpus has a slightly different eye movement feature set than what we have for the Dundee corpus. Therefore we treat the two eye movement corpora as two individual modalities in order to assess their individual contributions. GECO has 34 features reflecting word-based processing. Dundee has 30 word-based features that were extracted from the raw data and previously used for POS induction by Barrett et al. (2016). For GECO, we use the features that are already extracted by the authors of the corpus. Both corpora include five word-based features e.g., first fixation duration (which is a measure said to reflect early syntactic and semantic integration), total fixation time and fixation probability. The Dundee Corpus has more features concerning the context words whereas GECO has pupil size and many features distinguishing the different passes over a word.

3.3 Prosody

The prosody features are described in detail in Frermann and Frank (2017) and are freely available.[2] They are derived from the Brent (Brent and Siskind, 2001) and Providence (Demuth et al., 2006) portions of the CHILDES corpus (MacWhinney, 2000), comprising longitudinal datasets of raw speech directed to 22 children, and its transcription. Word-level speech-text alignments were obtained automatically using forced alignment. For each token-level audio snippet, a set of 88 prosody features was extracted based on a previously established feature set (Eyben et al., 2016), including standard features derived from F0–F3 formants, spectral shape and rhythm features, intensity and MFCC features among others. Type-level prosody features were obtained as averaged token-level features for each word type.

3.4 Keystroke features

We extracted keystroke features from the publicly available data from Killourhy and Maxion (2012). This data contains key hold times and pauses of all key presses of 20 subjects as they completed transcription and free composition tasks. We only used data from the free composition part. A pause is defined by the authors as the duration from keydown to keydown. The free composition data consists of a total of 14890 typed words and 2198 word types.

For each word, we extracted the following features: (i) average key hold duration of all characters associated with producing the word, (ii) pre-word pause, (iii) hold duration of space key before word, (iv) pause length of space key press pause before word, and (v) ratio of keypresses used in the word production to length of the final word. For each word, we also included these five features for up to 3 words before. In total, we have $5 * 4 = 20$ keystroke features. We use lower-cased word type averages, as with the other modalities.

3.5 Eigenwords

Eigenwords are standard, pre-trained word embeddings, induced using spectral-learning techniques (Dhillon et al., 2015). We used the 30-dimensional, pre-trained eigenvectors.[3]

3.6 Preliminary evaluation

Our application of these word representations and their combinations is unsupervised POS and syntactic chunk induction, but before presenting our projection methods in §4 and our experiments in §5, we present a preliminary evaluation of the different modalities using word association norms.

Table 1 shows the weighted correlation between cosine distances in the representations and the human ratings in the word association norm datasets available at `wordvectors.org` (Faruqui and Dyer, 2014a). Eigenwords, not surprisingly, correlates better than the representation based on processing data – with the exception of prosody. The correlation with prosody is non-significant, however, because of the small sample size.

4 Combining datasets

We now have word representations from different, complementary modalities, with very different coverages, but all including a small overlap. We assume that the different modalities contain complementary human text processing traces because they reflect different cognitive processes, which motivates us to combine these different sources of information. Our assumption is confirmed in the evaluation. The fact that we have very low coverage for some modalities, and the

[2] `https://github.com/ColiLea/prosodyAOA`

[3] `http://www.cis.upenn.edu/~ungar/eigenwords/`

fact that we have an overlap between all our vocabularies, specifically motivates an approach, in which we use the intersection of word types to learn a projection from two or more of these modalities into a shared space. Obviously, we can also simply concatenate our representations, but because of the low coverage of some modalities and because co-projecting modalities has some regularization effect, we hypothesize that it is better to learn a projection into a shared space. This hypothesis is verified by the results in §6.

4.1 Concatenating modalities

The simplest way of combining the modalities is concatenating the corresponding vectors for each word. The different modalities have different dimensionalities, so we would need to perform dimensionality reduction to sum or average vectors, and the non-overlapping words don't allow for e.g. taking the outer product, so we simply concatenate the vectors instead. We use 0 for missing values.

4.2 CCA

§4.2 and §4.3 describe two different projection methods for projecting the representations in the different modalities into a shared space. We use the intersection of the lower-cased vocabulary for the alignment, i.e., as a supervision signal. For example, if the words *man*, *dog* and *speak* exist in both eigenword and keystroke data, from these 2 x 3 vectors, CCA estimate the transformation for the vectors for *house*, *cat* and *boy*, which (in this example) only exists in the keystroke data.

Canonical Correlation Analysis (CCA), as originally proposed by Hotelling (1936), is a method of finding the optimum linear combination between two sets of variables, so the set of variables are transformed onto a projected space while the correlation is maximized. We use the implementation of Faruqui and Dyer (2014b) made for creating bilingual embeddings. We use modalities instead of languages. The size of the projected space is smaller than or equal to the original dimension.

We incrementally combine modalities and project them to new, shared spaces using the intersection of the lower-cased vocabulary. We add them by the order of word type count starting with the modality with most word types. For the first projection only, we reduce the size of the projected space. We set the ratio of the first projected space (only two modalities) to 0.6 based on POS induction results on development data using the setup described in §5.

4.3 SVD and Inverted Softmax

As an alternative to CCA, but closely related, we also use a projection method proposed and implemented by Smith et al. (2017), which uses singular value decomposition and inverted softmax (SVD+IS). This method uses a reference space, rather than projecting all modalities into a new space. Smith et al. (2017) apply SVD+IS to obtain an orthogonal transformation matrix that maps the source language into the target language. In addition, in order to estimate their confidence on the predicted target, they use an inverted softmax function for determining the probability that a target word translates back into a source word.

Like for CCA, we incrementally project datasets onto each other starting with the most word-type rich modality. We use the highest dimensionality of any of our representations (88 dimensions).

5 Experiments

This section presents our POS and syntactic chunk induction experiments. We present the datasets we used in our experiments, the sequence tagging architecture, based on second-order hidden Markov models, as well as the dictionary we used to constrain inference at training and test time.

5.1 Data

For unsupervised POS induction, we use Ontonotes 5.0 (Weischedel et al., 2013) for training, development and test. We set all hyper-parameters on the newswire (NW) domain, optimizing performance on the development set. Size of the development set is 154,146 tokens. We run individual experiments on each of the seven domains, with these hyper-parameters, reporting performance on the relevant test set. The domains are broadcast conversation (BC), broadcast news (BN), magazines (MZ), newswire (NW), the Bible (PT), telephone conversations (TC), and weblogs (WB). We also train and test unsupervised POS induction on the CoNLL 2007 (Nivre et al., 2007) splits of the Penn Treebank (Marcus et al., 1993) using the hyper-parameter settings from Ontonotes. We mapped all POS labels to Google's coarse-grained, universal POS tagset (Petrov et al., 2012). For model selection, we select based both on best results on Ontonotes

Rules		
DET	→	NP
VERB	→	VP
NOUN|PRONOUN|NUM	→	NP
.	→	O
ADJ	→	NP|ADJP
ADV	→	NP|VP|ADVP|AD
PRT	→	NP|PRT
CONJ	→	O|NP
ADP	→	PP|VP|SBAR

Table 2: Heuristics for expanding our POS dictionary to chunks

NW development as well as Penn Treebank development sets.

For syntactic chunk induction, we use the bracketing data from Penn Treebank with the standard splits for syntactic chunking. We tune hyperparameters for chunking on the development set and select best models based on the development result.

5.2 Model

We used a modification of the implementation of a type-constrained, second-order hidden Markov model with maximum entropy emissions from Li et al. (2012) (SHMM-ME). It is a second-order version of the first order maximum entropy HMM presented in (Berg-Kirkpatrick et al., 2010) with the important addition that it is constrained by a crowd-sourced tag dictionary (Wiktionary). This means that for all words in the Wiktionary, the model is only allowed to predict one of the tags listed for it in Wiktionary

The same model was used in Barrett et al. (2016) to improve unsupervised POS inducing using gaze data from the Dundee Corpus, and in Bingel et al. (2016) to augment an unsupervised POS tagger with features from fMRI recordings.

The number of EM iterations used for inducing our taggers was tuned using eigenvector embeddings on the development data, considering values 1..50. PoS performance peaked at iterations 30 and 31. We use 30 in all our POS experiments. For syntactic chunking, we use 48 iterations, which led to the best performance on the PTB development data using only eigenword embeddings.

5.3 Wiktionary

The Wiktionary constrains the predicted tags in our model. The better the Wiktionary, the better the predictions.

For POS-tagging we used the same Wiktionary dump[4] that Li et al. (2012) used in their original experiments. The Wiktionary dump associated word types with Google's universal parts-of-speech labels.

Feature set	TA
No embeddings	60.32
Eigenwords	59.26
Best combined models	
CCA Dun_GECO_Pros	**63.33***†
SVD+IS GECO_Key_Pros	62.91*
Concat Eig_GECO_Key	61.16

Table 3: Chunk tagging accuracy. Best models from CCA, SVD+IS and concatenation. Model section on development set. * $p < .001$ Mcnemar mid-p test when comparing to no embeddings. † $p < .001$ Mcnemar mid-p test when comparing to Eigenwords.)

For chunking, Wiktionary does not provide direct information about the possible labels of words. We instead apply simple heuristics to relate POS information to syntactic chunking labels. Since we already know the relation between words and POS labels from Wiktionary, we can compute the transitive closure in order to obtain a dictionary relating words with syntactic chunking labels. We present the heuristics in Table 2.

Note that the rules are rather simple. We do not claim this is the best possible mapping. We are relying on these simple heuristics only to show that it is possible to learn syntactic chunkers in an unsupervised fashion by relying on a combination of features from different modalities and a standard, crowd-sourced dictionary.

6 Results

All our POS tagging accuracies can be seen in Table 4. Our first observation is that human processing data helps unsupervised POS induction. In fact, the models augmented with processing data are *consistently* better than the baseline without vector representations, as well as better than only using distributional word embeddings.

Generally, CCA seems to find the best projection into a common space for system combinations. For Penn Treebank, the CCA-aligned model is the best and this result is significant ($p <$

[4] https://code.google.com/archive/p/wikily-supervised-pos-tagger/

	Ontonotes								PTB
Feature set	BC	BN	MZ	NW	PT	TC	WB	avg	
No embeddings	83.1	84.41	85.32	84.94	85.14	77.8	85.93	83.81	82.83
Eigenwords	83.16	84.68*	85.48	85.07	85.31	78.07	85.88	83.95	83.38*
Best Ontonotes NW models									
CCA Eig_Dun	**83.45***†	**84.99***	85.79*	**85.38***†	85.2	77.99	**86.38***†	84.17	**84.28***†
SVD+IS Dun_GECO_Key	83.24	84.76	**86.22***†	85.33*†	85.44	77.84	85.95	84.11	84.25*†
Concat Eig_Dun_GECO	83.39*†	84.78*	85.8*†	85.36*†	**85.45**	**78.38***	86.21†	**84.19**	83.91*†
Best PTB models									
CCA Eig_Dun	**83.45***†	**84.99***	85.79*	**85.38***†	85.2	77.99	**86.38***†	84.17	**84.28***†
SVD+IS Dun_Key	83.24	84.59	86.12*†	85.28*†	85.39	77.90	85.86	84.05	84.24*†
Concat Eig_Pros	83.22	84.54	85.67	85.01	84.98	77.98	85.97	83.91	84.22*†

Table 4: POS tagging accuracies for baselines and the model combinations that performed best on newswire development data (NW). Best performance per domain is boldfaced. *) $p < .001$ McNemar mid-p test when compared to the no embeddings condition for the corresponding test set. †) $p < .001$ McNemar mid-p test when compared to eigenwords for the corresponding test set.

.001) when comparing both to no embeddings and eigenwords. For Ontonotes 5.0, CCA is better than the other projection methods in 4/7 domains, but when averaging, concatenation gets the higher result.

The standard embeddings are often part of the best combinations, but the human processing data contributes with important information; in 4/7 domains as well as on PTB data, we see a significantly better performance ($p < .001$) with a combination of modalities when comparing to eigenwords.

Aligning Dundee with eigenwords is the best POS model both according to the Ontonotes 5.0 NW development set and the Penn Treebank development set. Dundee is the most frequent modality in the six best POS induction models with five appearances. Eigenwords is second most frequent with four appearances.

The syntactic chunking accuracies are in Table 3. Also here CCA is the better combination method. For chunking, all combined models are better than no embeddings and eigenwords. The improvement is significant compared to no embeddings for concatenation $p < .001$. For CCA, the result is significantly better than no embeddings and eigenwords.

For chunking, GECO data appears in all best models and is thus the most frequent modalities. Keystroke and prosody appears in two best models each.

	Keystroke	Dundee	GECO
Dundee	16.84		
GECO	11.39	1.02	
CCA all	13.98	3.72	3.09

Table 5: Graph similarities in $[0, \infty)$, $0 =$ identical.

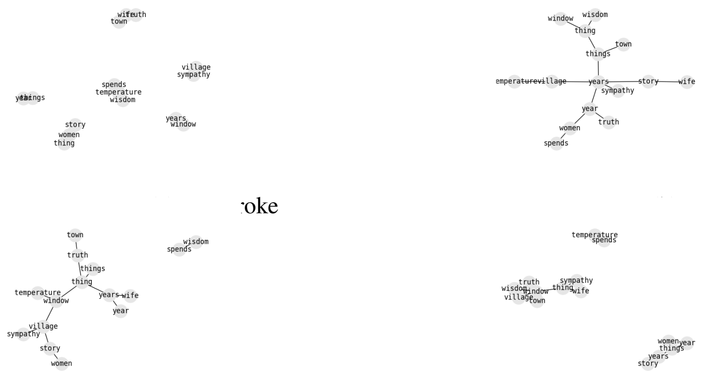
7 Analysis

7.1 What is in the vectors?

Nearest neighbor graphs We include a detailed analysis of subgraphs of the nearest neighbor graphs in the embedding spaces of keystrokes, Dundee, GECO, and CCA projection of all modalities. Specifically, we consider the nearest neighbor graphs among the 15 most frequent unambigous nouns, according to Wiktionary.[5] See Figure 2 for plots of the nearest neighbor graphs. The prosody features containing less than 600 word types only contained 2 of the 15 nouns and is therefore not included in this analysis.

Projecting word representations into a shared space using linear methods assumes approximate isomorphism between the embedding spaces - or at least their nearest neighbor graphs. We use the VF2 algorithm (Cordella et al., 2001) to verify that the subgraphs are *not* isomorphic, but this can also be seen directly from Figure 2. Neither keystroke and gaze embeddings, nor the two different gaze-induced embeddings are isomorphic.

[5] Wiktionary is a crowd-sourced, imperfect dictionary, and one of the ”unambiguous nouns” is *spends*, which, we assume, you are more likely to encounter as a verb.



·oke

lee

(c) Geco

(d) CCA projection, all modalities

Figure 2: Nearest neighbor graphs for 15 frequent nouns.

Since none of the modalities induce isomorphic nearest neighbor graphs, this does not tell us much about similarities between modalities. To quantify the similarity of non-isomorphic graphs, we use *eigenvector similarity* Shigehalli and Shettar (2011), which we calculate by computing the Laplacian eigenvalues for the nearest neighbors, and for each graph, find the smallest k such that the sum of the k largest eigenvalues is $<90\%$ of the eigenvalues. We then take the smallest k of the two, and use the sum of the squared differences between the largest k eigenvalues as our similarity metric.

Using this metric to quantify graph similarity, we see in Table 5 that, not surprisingly, the gaze graphs are the most similar. The projected space is more similar to the gaze spaces, but balances gaze and keystroke information. The GECO embeddings agree more with the keystrokes than the Dundee embeddings does.

t-SNE plots We take words that–according to the Wiktionary–can only have one tag and sort them by BNC frequency (Kilgarriff, 1995) in descending order. For these words and their POS tags we get the feature vector of the POS model yielding the highest result on both Ontonotes and PTB: CCA-projected eigenwords and Dundee features. For the first 200 occurrences of the frequency-sorted list, we reduce dimensionality using t-Distributed Stochastic Neighbor Embedding (t-SNE) (Maaten and Hinton, 2008) and plot the result. Figure 3 shows that 200 most frequent content words cluster with respect to their POS tag, somewhat distinguishing verbs from nouns and adjectives from adverbs in CCA space.

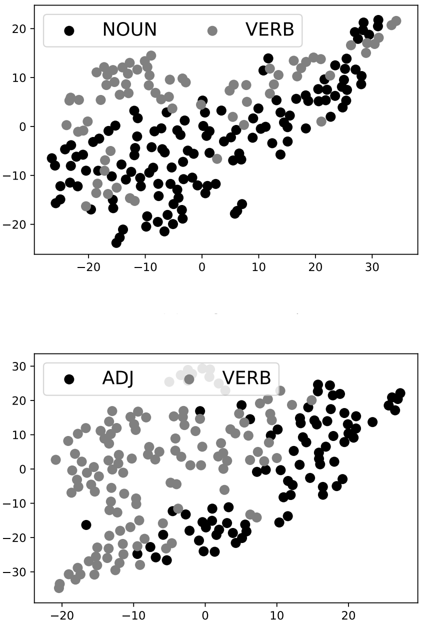
7.2 How big a Wiktionary do we need?

Our Wiktionary for English contains POS information for 72,817 word types. Word types have 6.2 possible POS categories on average meaning we have over 450.000 entries in our POS dictionary. For Penn Treebank, 70.0% of wordtypes of the test set are covered by the dictionary. For the chunking data, 70.4% of wordtypes of the test set are covered by the dictionary. The English Wiktionary is thus much bigger than wiktionaries for low-resource language (Garrette and Baldridge, 2013). How big a dictionary is needed to achieve good performance, and can we get away with a smaller dictionary if we have processing data? This section explores the performance of the model as a function of the Wiktionary size.

We sorted the Wiktionary by word frequency obtained from BNC (Kilgarriff, 1995) and increased the Wiktionary size for the best POS system starting with 0 (no dictionary). For each Wiktionary size, we compare with the baseline without access to processing data and eigenwords. The learning curve can be seen in Figure 4a and Figure 4b. We observe that having entries for the most frequent words is a lot better than having no dictionary, and that the difference between our best system and the baseline exists across all dictionary sizes. With 10,000 entries, all systems seems to reach a plateau.

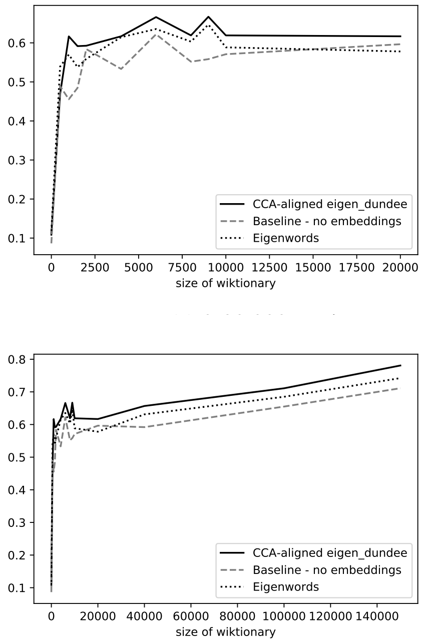
8 Discussion

Genres and domains When collecting our human language processing data, we did not control for genre. Our data sets span child-directed speech, free text composition, and skilled adults reading fiction and newspaper articles. The



(b) ADJ and VERB

Figure 3: t-SNE plots of CCA-projected eigen_dundee features for pairs of tags.



(b) 0-150,000 entries

Figure 4: Learning curve assuming Wiktionary entries for k most frequent words, comparing our best PoS induction system against our baseline. On Ontonotes WB development data, 30 training iterations.

Dundee corpus (newspaper articles) matches the genre of at least some of the Ontonotes test set. Immonen and Mäkisalo (2010) found that for keystroke, genre does seem to have an effect on average pause length, be it sentence initial, word initial, clause initial or phrase initial. Texts organized linearly–e.g. reports and narratives–require less pausing than texts with a global approach, like expository, persuading and generalizing text. Our results show that human processing features transfer across genres, but within-genre data would probably be beneficial for results.

Richer representations The type-level features we use, do not take context into account, and the datasets we use, are too small to enrich our representations. Human processing data is more and more readily available, however. Eye trackers are probably built into the next generation of consumer hardware, and speech records and keystroke logs are recordable with existing technology.

9 Conclusion

We have shown how to improve unsupervised POS induction and syntactic chunking significantly using data reflecting human language processing. Our model, which is a second-order hidden Markov model, is the first to combine multidimensional, continuous features of eye movements, prosody and keystroke logs. We have shown that these features can be combined using projection techniques, even when they only partially overlap in word coverage. None of our models require access to these features at test time. We experimented with all combinations of modalities, and our results indicate that eye tracking is useful for both chunking and POS induction. Finally, we have shown that the potential impact of human processing data also applies in a low-resource setting, i.e., when available tag dictionaries are small.

Acknowledgements

Thanks to Desmond Elliott for valuable ideas.

This research was partially funded by the ERC Starting Grant LOWLANDS No. 313695, as well as by Trygfonden.

References

Veerle M Baaijen, David Galbraith, and Kees de Glopper. 2012. Keystroke analysis: Reflections on procedures and measures. *Written Communication* 29(3):246–277.

Maria Barrett, Željko Agić, and Anders Søgaard. 2015. The Dundee treebank. In *The 14th International Workshop on Treebanks and Linguistic Theories (TLT 14)*. pages 242–248.

Maria Barrett, Joachim Bingel, Frank Keller, and Anders Søgaard. 2016. Weakly supervised part-of-speech tagging using eye-tracking data. In *ACL*. pages 579–584.

Maria Barrett and Anders Søgaard. 2015a. Reading behavior predicts syntactic categories. *CoNLL 2015* pages 345–349.

Maria Barrett and Anders Søgaard. 2015b. Using reading behavior to predict grammatical functions. In *Workshop on Cognitive Aspects of Computational Language Learning (CogACLL)*. pages 1–5.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Cote, John DeNero, , and Dan Klein. 2010. Painless unsupervised learning with features. In *NAACL*. pages 582–590.

Joachim Bingel, Maria Barrett, and Anders Søgaard. 2016. Extracting token-level signals of syntactic processing from fmri-with an application to pos induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 747–755.

M. R. Brent and J. M Siskind. 2001. The role of exposure to isolated words in early vocabulary development. *Cognition* 81:31–44.

Patricia A Carpenter and Marcel Adam Just. 1983. What your eyes do while your mind is reading. *Eye movements in reading: Perceptual and language processes* pages 275–307.

Uschi Cop, Nicolas Dirix, Denis Drieghe, and Wouter Duyck. 2017. Presenting geco: An eyetracking corpus of monolingual and bilingual sentence reading. *Behavior research methods* 49(2):602–615.

L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. 2001. An Improved Algorithm for Matching Large Graphs. *Proc. of the 3rd IAPR TC-15 Workshop on Graphbased Representations in Pattern Recognition* 17:1–35.

Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition* 109(2):193–210.

Katherine Demuth, Jennifer Culbertson, and Jennifer Alter. 2006. Word-minimality, epenthesis and coda licensing in the early acquisition of english. *Language and Speech* 49(2):137–173.

Paramveer Dhillon, Dean Foster, and Lyle Ungar. 2015. Eigenwords: Spectral word embeddings. *Journal of Machine Learning Research* 16:3035–3078.

Florian Eyben, Klaus Scherer, Bjrn Schuller, Johan Sundberg, Elisabeth Andr, Carlos Busso, Laurence Devillers, Julien Epps, Petri Laukka, Shrikanth Narayanan, and Khiet Phuong Truong. 2016. The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing 7(2):190–202.

Manaal Faruqui and Chris Dyer. 2014a. Community evaluation and exchange of word vectors at wordvectors.org. In *ACL: System Demonstrations*. pages 19–24.

Manaal Faruqui and Chris Dyer. 2014b. Improving vector space word representations using multilingual correlation. In *EACL*. pages 462–471.

Lea Frermann and Michael C. Frank. 2017. Prosodic features from large corpora of child-directed speech as predictors of the age of acquisition of words. *CoRR* .

Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. In *NAACL*.

Adam Goodkind and Andrew Rosenberg. 2015. Muddying the multiword expression waters: How cognitive demand affects multiword expression production. In *MWE@ NAACL-HLT*. pages 87–95.

Sini Immonen and Jukka Mäkisalo. 2010. Pauses reflecting the processing of syntactic units in monolingual text production and translation. *HERMES-Journal of Language and Communication in Business* 23(44):45–61.

Alan Kennedy, Robin Hill, and Joël Pynte. 2003. The Dundee Corpus. In *Poster presented at ECEM12: 12th European Conference on Eye Movements*.

Alan Kennedy and Joël Pynte. 2005. Parafoveal-on-foveal effects in normal reading. *Vision research* 45(2):153–168.

Adam Kilgarriff. 1995. BNC database and word frequency lists. *Retrieved Dec. 2017* .

Kevin S Killourhy and Roy A Maxion. 2012. Free vs. transcribed text for keystroke-dynamics evaluations. In *Proceedings of the 2012 Workshop on Learning from Authoritative Security Experiment Results*. ACM, pages 1–8.

Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. *NAACL* pages 1528–1533.

Shen Li, João Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *EMNLP*. pages 1389–1398.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research* 9(Nov):2579–2605.

Brian MacWhinney. 2000. *The CHILDES project: Tools for analyzing talk.*. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, third edition edition.

Mitchell Marcus, Mary Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics* 19(2):313–330.

Ann Matsuhashi. 1981. Pausing and planning: The tempo of written discourse production. *Research in the Teaching of English* pages 113–134.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser. *Natural Language Engineering* 13(2):95–135.

John K Pate and Sharon Goldwater. 2011. Unsupervised syntactic chunking with acoustic cues: computational models for prosodic bootstrapping. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*. Association for Computational Linguistics, pages 20–29.

John K Pate and Sharon Goldwater. 2013. Unsupervised dependency parsing with acoustic cues. *Transactions of the Association for Computational Linguistics* 1:63–74.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *LREC*. pages 2089–2094.

Barbara Plank. 2016. Keystroke dynamics as signal for shallow syntactic parsing. In *COLING*. pages 609–618.

Keith Rayner. 1998. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin* 124(3):372–422.

Keith Rayner and Susan A Duffy. 1988. On-line comprehension processes and eye movements in reading. *Reading research: Advances in theory and practice* 6:13–66.

Vijayalaxmi Shigehalli and Vidya Shettar. 2011. Spectral Technique using Normalized Adjacency Matrices for Graph Matching. *International Journal of Computational Science and Mathematics* 3:371–378.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium* .