Bridging the gap between speech technology and natural language processing: an evaluation toolbox for term discovery systems

Bogdan Ludusan*, Maarten Versteegh*, Aren Jansen†, Guillaume Gravier‡, Xuan-Nga Cao*
Mark Johnson§, Emmanuel Dupoux*

* LSCP - EHESS/ENS/CNRS, Paris; † CLSP - Johns Hopkins University, Baltimore
‡ IRISA - CNRS, Rennes; § Macquarie University, Sydney

Abstract

The unsupervised discovery of linguistic terms from either continuous phoneme transcriptions or from raw speech has seen an increasing interest in the past years both from a theoretical and a practical standpoint. Yet, there exists no common accepted evaluation method for the systems performing term discovery. Here, we propose such an evaluation toolbox, drawing ideas from both speech technology and natural language processing. We first transform the speech-based output into a symbolic representation and compute five types of evaluation metrics on this representation: the quality of acoustic matching, the quality of the clusters found, and the quality of the alignment with real words (type, token, and boundary scores). We tested our approach on two term discovery systems taking speech as input, and one using symbolic input. The latter was run using both the gold transcription and a transcription obtained from an automatic speech recognizer, in order to simulate the case when only imperfect symbolic information is available. The results obtained are analysed through the use of the proposed evaluation metrics and the implications of these metrics are discussed.

Keywords: evaluation, spoken term discovery, word segmentation

1. Introduction

Unsupervised discovery of linguistic structures is attracting a lot of attention. Under the so-called 'zero resource setting' (Glass, 2012), a learner has to infer linguistic units from raw data without having access to any linguistic labels (phonemes, syllables, words, etc.). This can have applications in languages with little or no resources, and has considerable relevance for cognitive modelling of human infants language acquisition (Jansen et al., 2013).

One area of particular interest is the automatic discovery of words or 'terms' from unsegmented input. This particular problem has been addressed from the viewpoint of at least two language processing communities: natural language processing (NLP) and speech technology (ST). Most of the systems from the NLP community take as input a speech corpus that has been transcribed phonemically (gold transcription), but where the word boundaries have been deleted (Brent and Cartwright, 1996; Brent, 1999; Johnson et al., 2007; Goldwater et al., 2009). The aim is to recover these boundaries, as well as to construct a lexicon of terms. Note that most of these algorithms exhaustively 'parse' their inputs in terms of a sequence of word tokens. A set of standard evaluation criteria has been established: segmentation, word token and word type precision, recall and F-scores. The corpora are for the most part in English (Daland and Pierrehumbert, 2011), although a small number of studies are now conducted across different languages (Fourtassi et al., 2013; Daland and Zuraw, 2013). The algorithms of term discovery coming out of the ST community also attempt to discover terms, but work from the raw speech input, and may not produce an exhaustive parse. These systems are more recent and have not yet converged on an accepted set of corpora and evaluation methods (Park and Glass, 2008; Jansen and Van Durme, 2011; Flamary et al., 2011; McInnes and Goldwater, 2011; Muscariello et al., 2012). The name term discovery (TD) will be used throughout this paper for both kinds of systems.

The aim of this paper is to propose both a corpus as well as a set of evaluation tests that would enable researchers to compare the performance of different systems within and across communities. As new ST/NLP hybrid systems are emerging (Lee and Glass, 2012), it is our belief that a common evaluation method will be useful to bridge the gap between the two communities.

2. Evaluation method

Algorithms for discovering recurring patterns in linguistic data can be used for a variety of purposes: speech corpora indexing, keyword search, topic classification, etc. We do not claim that a single evaluation method is relevant for all these applications. Rather, we propose a toolbox containing several evaluation metrics, each one tailored to measure a different subcomponent of the TD algorithm. The reason for proposing such a toolbox, rather than a single measure, is that it enables more fine grained comparisons between systems. In addition, it enables, a system diagnostic tool to assess which subcomponent needs improvement. Another design feature of our evaluation toolbox is that it is performed in the phoneme space, i.e., aligning the waveform with gold phonemic transcription. This is a useful feature to enable a comparison of ST and NLP systems.

Extracting recurring terms from continuous speech is a problem that involves several interconnected components (see Figure 1). Firstly, one component involves *matching* stretches of speech input. This is typically done with a Dynamic Time Warping technique (DTW), and can be viewed as constructing a list of pairs of fragments, each corresponding to stretches of speech. We propose in this paper several methods for evaluating matching quality. Secondly, many systems also incorporate a mechanism for *clustering* fragments into candidate terms. Some systems memorize these clusters in a library and use them for extracting further

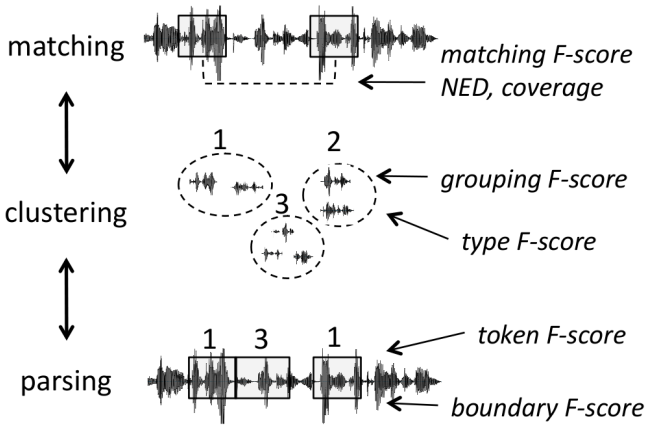

Figure 1: Logical pipeline highlighting three components that can be part of term discovery systems, and presentation of our 5-levels evaluation toolbox. The top two (matching and grouping scores) use the aligned phoneme transcription as gold standard, and the last three (type, token and boundary scores) use the word level alignment.

fragments (Muscariello et al., 2012), others perform the fragment clustering only as the last step (Park and Glass, 2008; Jansen and Van Durme, 2011; Flamary et al., 2011; McInnes and Goldwater, 2011). Clustering quality can be evaluated rather standardly in terms of the purity/inverse-purity of their phonemic content. Thirdly, the extracted clusters or fragments are used for *parsing* the input and assign segmentation boundaries. Some systems perform parsing implicitly (as a trace of the matching process), others, perform an explicit parsing step, allowing to clean up potentially overlapping matches. The discovered clusters and the parses can be evaluated in terms of a gold lexicon and a gold alignment. For this, we use the standard NLP metrics (*type*, *token* and *boundary* F-score).

Note, however, that contrary to NLP systems, most ST systems do not exhaustively parse their input. It is therefore important to compute the NLP type statistics on the part of the corpus that has been covered, while keeping track of a separate coverage statistic. In contrast to ST systems, NLP systems do not work from raw speech. In order to compare them, we therefore complement the word segmentation NLP systems with a speech recognition front-end, and perform the evaluation on the entire front-end plus the word segmentation pipeline.

2.1. Precision, recall and F-score

We use the same logic at all the defined levels of the toolbox, i.e., we define a set of found structures (X), which we compare to the set of gold structures (Y) using average precision, recall and F scores as defined in (1). In most of the cases, X and Y will be sets of fragments (i, j) or of pairs of such fragments. We will always sum over fragment types, as defined through their phonemic transcriptions T, with a weight w defined as the normalized frequency of the types in the corpus. The function $match(t, X)$ counts how many tokens of type t are in the set X.

$$Precision = \sum_{t \in types(X)} w(t, X) \times \frac{match(t, X \cap Y)}{match(t, X)}$$

$$Recall = \sum_{t \in types(X)} w(t, X) \times \frac{match(t, X \cap Y)}{match(t, Y)}$$

$$F\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (1)$$

2.2. Alignment

Previous evaluation techniques of ST-TD systems have used the word as the level of alignment (e.g. Park and Glass (2008), Muscariello et al. (2012)). However, in order to obtain a more fine grained evaluation, and to enable a comparison with NLP systems, we align the signal with phoneme-level annotations. As the speech part of the system has no knowledge of the segmental content in the signal it processes, a discovered boundary may fall between two annotated phoneme boundaries. In order to transcribe a given fragment, we consider as being part of the annotation any phoneme that has either at least 50% overlap in time with the fragment, or at least 30ms overlap. By setting a 30 ms overlap we impose a minimum limit for a chunk to be perceived as belonging to a certain category (30ms being arguably the upper bound of the minimum amount of speech needed to identify a phoneme (Tekieli and Cullinan, 1979)), while using the 50% limit we take into consideration also short phonemes, if there is sufficient overlap with the said fragment. Note that, through the alignment, the representation level for the system evaluation has changed: Matches found at the acoustic level are evaluated at the phonemic level. Thus, each found acoustic fragment is treated like a separate phoneme string occurrence during the evaluation.

2.3. Matching quality

We propose two sets of measures of matching quality, one qualitative and easy to compute, and the other quantitative and computationally intensive. With respect to the former type, we propose the normalized edit distance (NED) and the coverage. The NED is the Levenstein distance between each two string occurrences, divided by the maximum of the length of the two strings. It expresses the goodness of the pattern matching process and can be interpreted as the percentage of phonemes shared by the two strings. The coverage is defined by the percentage of phonemes corresponding to discovered fragments from the total number of phonemes in the corpus. These two values give an intuitive idea of the matching quality and can capture the trade-off between very selective matching algorithms (low NED, low coverage), and very permissive ones (high NED, high coverage). A formal definition of these measures is presented in Equations 2 and 3, where P_{disc} is the set of discovered fragment pairs, and P_{gold} the gold set of non-overlapping phoneme-identical pairs. Note that coverage is computed over all found fragments (F_{disc}), some of which may be hapaxes (clusters of size 1) and therefore do not occur in pairs.

$$NED = \sum_{(x,y)\in P_{disc}} \frac{ned(x,y)}{|P_{disc}|} \quad (2)$$
$$ned((i,j),(k,l)) = \frac{Levenstein(T_{i,j}, T_{k,l})}{max(j-1+1, k-l+1)}$$

$$Coverage = \frac{cover(F_{disc})}{cover(F_{gold})} \quad (3)$$
$$cover(P) = \cup_{(i,j)\in P}\{i, i+1, ..., j\}$$

A more quantitative evaluation is given by the precision, recall and F-scores of the set of discovered pairs with respect to all possible matching pairs of substrings in the corpus. For efficiency, we restrict the substrings to a particular range, e.g., between 3 and 30 phonemes. Precision is computed as the proportion of discovered substring pairs which belong to the list of gold pairs. In order for this statistic not to be dominated by very frequent substrings (the number of pairs grows with the square of the frequency), we compute these proportions across pairs of the same type, re-weighted by the frequency of the type. Note that, as the gold list contains all of the substrings, we have to augment the discovered pair set with all the logically implied substrings. The proper way to generate those would be to read them off the DTW alignment. However, this information is not accessible in most ST-TD systems. We therefore re-generate the list of discovered substrings using DTW in phoneme space. This allows not to penalize too much an algorithm discovering the pair *democracy/emocracy*; indeed, this match will generate the correct substring match *emocracy/emocracy*, and many other smaller ones. By a similar computation, we can define recall as being the proportion of gold pairs present in the discovered set. For systems not separating matching from clustering, the matching quality can be still computed by decomposing the found clusters into a list of matching pairs and applying the above algorithm. The measures are defined formally in Equation 1, while X is the substring completion of P_{disc}, Y is the set of all non overlapping matching substrings in the gold transcript of minimum length 3, and the functions involved in their computation are defined in the following equations.

$$types(X) = \{T_{i,j},\, where\, (i,j) \in flat(P)\} \quad (4)$$
$$w(t,X) = \frac{freq(t,X)}{|flat(X)|}$$
$$match(t,X) = |\{(x,(i,j)) \in X,\, where\, T_{i,j} = t\}|$$
$$freq(t,X) = |\{(i,j) \in flat(X),\, where\, T_{i,j} = t\}|$$
$$flat(X) = \{(i,j),\, where\, \exists x\, (x,(i,j)) = t \in X\}$$

2.4. Grouping quality

We propose to compute grouping quality using the pairwise matching approach as above (see also Amigo et al. (2009)) but not expanding the set of pairs using substrings, and restricting the analysis to the covered corpus. Apart from that, the computation is the same as above, i.e. averaging across all matching pairs of the same types and re-weighting by type frequency. The interpretation of Grouping quality is different from that of Matching quality. Matching quality is asking how well the algorithm is able to locate any identical stretches of speech in the whole corpus. Grouping quality is asking how good and homogeneous the discovered groups of fragments are. Again, the measures used here are defined in Equation 1, while the sets involved in their computation are defined in 5. As in the These sets are constructed as the sets of all pairs of fragments belonging to the same cluster.

$$\begin{aligned} X = \{&((i,j),(k,l))\, where\, \exists\, c \in C_{disc}\, (i,j) \in c \\ &and\, (k,l) \in c\} \\ Y = \{&((i,j),(k,l)) \in F_{all} \times F_{all},\, where \\ &\exists\, c_1, c_2 \in C_{disc}\, (i,j) \in c_1 and\, (k,l) \in c_2\, and \\ &T_{i,j} = T_{k,l}\, and\, \{i,...,j\} \cap \{k,...,l\} = \emptyset\} \end{aligned} \quad (5)$$

where C_{disc} is the set of discovered clusters, each cluster being a set of fragments.

2.5. Token, Type and Boundary quality

For systems that output clusters and use them to parse the input, it is possible to evaluate the relevance of these clusters with respect to a gold lexicon, and to the gold word boundaries. Here, we apply the exact same definitions as in NLP systems: for instance, token recall is defined as the probability that a gold word token has been found in some cluster (averaged across gold tokens), while token precision represents the probability that a discovered token matches a gold token (averaged across discovered tokens). The F-score is the harmonic mean between the two. Similar definitions are applied for the Type score. Again, the same formal definition of the metrics is employed here (Equation 1). The subsets involved in their computation are the following:

$$\begin{aligned} &X = F_{\text{disc}} : \text{set of discovered fragments} \\ &Y = \{(i,j) \in F_{all},\, where\, T_{i,j} \in L\, and \\ &\quad i,j \in cover(X)\} \end{aligned} \quad (6)$$

The $flat$ function in Equation 4 is being redefined as the identity function. The only difference between type and token scores, is that for type, the weighting function is redefined as a constant:

$$w(t,X) = \frac{1}{|types(F_{disc})|} \quad (7)$$

The Boundary score is defined in a different manner, to take into account the fact that the fragments found might not be perfectly aligned to the gold transcription. We use Equation 1, with sets and functions defined as follows:

$$\begin{aligned} &X = \text{set of discovered boundaries} \\ &Y = \text{gold set of boundaries} \\ &\text{there is only one type for the boundaries} \\ &match(t,X) = |X| \qquad w(t,X) = 1 \end{aligned} \quad (8)$$

As we said above, these statistics are be computed over the *covered* corpus, in order to have comparable statistics with NLP methods. For systems which do not return clusters, a standard graph clustering algorithm can be added and the same procedure followed.

3. System Presentation

Two ST systems were employed in this paper: the JHU system (Jansen and Van Durme, 2011), and MODIS (Catanese et al., 2013). We have chosen to use these systems in order to demonstrate the appropriateness of the proposed evaluation method for different ST-TD algorithms. On the NLP side, we used the Adaptor Grammar (AG) (Johnson et al., 2007) framework to perform word segmentation. Further details about these systems can be found in the following paragraphs.

3.1. JHU system

The first unsupervised word discovery approach we evaluated was the efficient segmental DTW-based system presented in (Jansen and Van Durme, 2011). The system operates in two stages. First, two randomized algorithms–locality sensitive hashing (LSH) and point location in equal balls (PLEB)–are used to enable linear-time computation of an approximate self-similarity matrix that covers the entire corpus. Second, efficient image processing techniques are used to search for syllable-sized repetitions that manifest themselves as short diagonal line segments in the similarity matrix. The center points of the these syllable repetitions are then used to seed a local segmental DTW search for the full extent of the repetition.

This core repetition discovery procedure produces a large collection of isolated segment pairs that contain similar lexical content. This must be post-processed into a single pseudo-word transcript by (i) applying a graph clustering procedure to group individual word repetitions into pseudo-word categories; (ii) constructing a pseudo-word lattice corresponding to each recording in the corpus; and (iii) performing a Viterbi decode of the lattice. The lattice arc confidence scores are derived from the DTW match similarity and a path from start to final nodes are guaranteed through the addition of epsilon arcs as needed. No language model scores are applied in the Viterbi search. The resulting automatic time aligned transcripts, which tend to be sparse due to incomplete pseudo-word coverage, are then provided to the evaluation tools described herein for scoring.

The JHU system includes all of the three components described in Figure 1, and applies them sequentially. The core repetition discovery performs *matching* and outputs a list of pairs. Graph clustering outputs a list of *clusters*. The Viterbi decoding performs *parsing*. This enables us to evaluate each of the successive outputs using our toolboxes, in order to evaluate the contribution of each component along this pipeline.

3.2. MODIS

MODIS[1] implements the motif discovery algorithm of (Muscariello et al., 2012). A segmental variant of a dynamic time warping (DTW) algorithm is used to search for a re-occurrence of a seed (a short segment of speech) in a portion of the data. When a match—i.e., a segment for which the distortion along the warping path is below a threshold—is found for the seed, an extension of the match is triggered to find the maximal length segment that remains below a similarity threshold α. Searching for a match of the seed is only performed within a time window right after the seed's location as in (Herley, 2006). A library of motifs is incrementally built, gathering all the motifs and their found occurrences. For each seed, a match is first sought in the library to account for long span repetitions. Search in the near future of the seed is only triggered if no matching motif was found in the library.

The parameters of MODIS are (a) the seed length (.25 s), (b) the size of the near future time window (90 s), and (c) the similarity threshold α. The lower the threshold, the more similar the occurrences of a motif within the library of motifs constructed. Here, it was empirically set by listening tests on a small fraction of the data ($\alpha = 4$). No post-processing clustering stage is implemented and silences were removed artificially before processing the data after concatenation of all the files. Minimum post-processing of the motif occurrence time was performed to deal with motifs spanning multiple non-silence segments.

Note that MODIS combines all of the three components described in Figure 1 simultaneously. The fragments stored in the library are used both to *match* and *parse*, and the outcome of the match result in an update of the library of *clusters*.

3.3. Adaptor Grammar

Adaptor Grammars (AG) is a framework for expressing hierarchical non-parametric Bayesian models as a kind of a probabilistic phrase structure grammar (Johnson et al., 2007). An AG system can learn recurring sub-trees, where the grammar specifies how these trees are organised. The state-of-the-art system on the Bernstein-Ratner corpus (Bernstein-Ratner, 1987) achieves close to 90% token f-score on the word segmentation task. It uses a complex grammar defining several levels above the words (collocations) and below the words (syllables and syllable fragments)(Johnson and Goldwater, 2009). Here, we will restrict ourselves to simpler grammars for the sake of the evaluation.

The simplest word segmentation Adaptor Grammar model is the *unigram model*: each utterance is generated as a sequence of words, where a word is a sequence of phones. Specifically, a unigram model has the following grammar rules:

unigram model	*monosyllable model*
$Utterance \rightarrow Word+$	$Utterance \rightarrow Word+$
$\underline{Word} \rightarrow Phone+$	$\underline{Word} \rightarrow Syllable$
$Phone \rightarrow ay\|b\|t\|...$	$Syllable \rightarrow (Onset)Rhyme$
	$Rhyme \rightarrow Nucleus(Coda)$
	$\underline{Onset} \rightarrow Consonant+$
	$\underline{Nucleus} \rightarrow Vowel+$
	$\underline{Coda} \rightarrow Consonant+$
	$Consonant \rightarrow b\|t\|p\|d\|...$
	$Vowel \rightarrow ay\|ey\|iy\|...$

The $Word$ non-terminal in the unigram grammar is *adapted*, which means that the model memorises or

[1] Freely available at http://gforge.inria.fr/projects/motifdiscovery or via a web service at http://allgo.irisa.fr

"learns" the sequences of phones that constitute a word. Our second AG model is the *monosyllable* model, which assumes that words are always monosyllabic (which is often true at the token level in a language like English). Here, parentheses indicate optionality (i.e., the $Onset$ and the $Coda$ are optional). The monosyllable AG adapts or learns the $Onset$, $Nucleus$ and $Coda$ non-terminals, as well as $Word$. That is, it learns the commonly occurring consonant sequences that occur at the beginnings and endings of words (i.e., onsets and codas respectively), as well as the words that are assembled out of these units.

These models only take discrete symbolic input. As a baseline, we test them with the *gold* transcriptions (in which case matching and grouping will be at ceiling performance). In order to compare these models with the ST systems, we also test them with the output of a phone recognizer (*reco*). In this case, the matching and grouping evaluations reflect the phone recognition part of the pipeline. AG models are built under the assumption that each word type is always pronounced in the same way; therefore, they can't capture this pronunciation variability and each variant of a word has to be stored as another type. We therefore expect a drop in performance in this setting. In order not to penalize too much the system, we took the best possible phone recognition performance we could get. We trained a triphone three-state HMM system with 17 Gaussian mixtures, 200 context-dependent triphones, a bigram language model, and speaker-specific re-adaptation. Because we wanted to achieve the best possible decoding, we trained the HMM and performed the decoding on the entire corpus. The phone accuracy was 59.2%.

The combined phoneme recognition + AG pipeline implements the three components in Figure 1 in the following way: The phone recognizer is responsible for matching, and AG performs both clustering and parsing simultaneously.

4. Materials

The proposed method was used to evaluate the systems on English, on the Buckeye corpus (Pitt et al., 2007). We have chosen this corpus for two reasons: first, for its characteristics and size. The Buckeye corpus contains around 38 hours of recordings which are entirely annotated at phonemic level, the annotation being needed during the evaluation process. Also, it contains conversational speech, which is more challenging than most types of speech used before for TD systems, and which is closer to the type of speech that people actually use in spoken interaction. This could encourage the development of more applications to be used in the day to day life. The second reason behind our choice was the fact that the corpus is free for non-commercial purposes. Thus, not using a proprietary corpus, would allow all research groups involved in the field to evaluate their system on the same dataset, using the same evaluation metrics. At the same time it might the boost the research in the field, by allowing other groups to come up with new approaches, once a standard evaluation framework is in place.

Note that the Buckeye has two levels of annotations: the word level, with citation form phoneme transcriptions, and the detailed phone transcription. As we need phoneme-level alignment for the evaluation, we used this latter transcription to setup our gold standard.

number of talkers	20 M, 20 F
duration	$\approx$ 38h
nb of utterances	$\approx$ 50k
nb of word types	$\approx$ 292k
nb of word tokens	$\approx$ 31k

Table 1: Buckeye corpus descriptive statistics

5. Experiments

We provided to all the systems the raw acoustic signal converted into standard 39 MFCC plus delta and delta-delta coefficients, which were mean-variance normalized for each file. Each original file from Buckeye was cut into smaller files at the onset and offset of every with silences, noises, or non-linguistic sounds which were removed. This resulted in 50k files which we consider as utterances (see Table 1 for more details. The output of each system was converted into a list of fragments defined as time onsets and offsets, as well as cluster number. The evaluations pipeline described above was then uniformly applied to the output of all systems[2]. Due to the very large number of fragment pairs involved in this corpus, we could not run the matching and grouping statistics on the entire corpus (these statistics scale to the square of the number of fragments, and generating DTW sub-fragments becomes a limiting factor). Rather, we performed 50 batches of random sub-sampling of 4% of the files, which was more tractable. The standard deviation was less than 10^{-5}; we computed the average precision and recall, and the F-scores from these averages.

6. Results and discussion

We first review the results of the AG pipelines. The AG pipelines with *gold* input shows good results on token, type and boundary scores on this rather large and variable corpus. This compares well with scores obtained on the Bernstein-Ratner corpus, given the nature of the corpus used in the experiments. For obvious reasons, the phoneme level scores (matching and grouping) are at their ceiling value. The lexical metrics used show the usual result that token statistics are better than type statistics, which corresponds to the fact that the adaptor grammar is frequency sensitive: it is better at segmenting high frequency words than low frequency ones. Finally, boundary recall is slightly higher than boundary precision, which suggests that the system has a tendency for over-segmenting. This is effect is stronger for the monosyllable grammar, which limits the size of the words and therefore tends to over-segment more than the unigram grammar which does not have a hard constraint on word size.

The performance of the AG pipelines on the $reco$ input drops considerably on all our metrics. Note that the low matching and grouping scores are to be uniquely attributable to the phone recognizer part of the pipeline,

[2]Due to a data conversion error, the AG system was tested on a slightly different fragment of the corpus than the JHU and MODIS systems; the two corpora overlapped by 96.4%.

	matching					grouping			token			type			boundary		
	NED	cov.	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
JHU																	
matching	85.7	35.8	0.2	16.6	0.3	3.4	26.4	5.7	0.6	1.9	0.9	0.6	0.7	0.6	19.3	14.9	16.9
clustering	39.1	28.1	0.1	13.5	0.2	4.9	72.5	9.1	0.8	1.1	0.9	0.6	0.5	0.6	27.5	8.5	13.0
parsing	50.7	24.5	2.7	29.6	4.9	4.9	8.3	6.0	0.3	0.2	0.3	0.2	0.2	0.2	34.0	5.5	9.5
MODIS	78.8	8.4	7.5	29.5	11.9	11.6	1.8	3.2	1.0	3.2	1.5	0.5	0.6	0.6	35.6	5.6	9.7
AG																	
unigram gold	0	100	100	100	100	100	100	100	54.1	68.5	60.5	30.9	41.3	35.4	82.9	84.8	83.2
monos gold	0	100	100	100	100	100	100	100	51.0	76.9	61.4	36.4	44.7	41.3	78.7	88.6	83.3
unigram reco	84.9	100	8.8	14.0	10.8	4.3	52.4	7.9	4.8	6.9	5.7	1.6	1.1	1.3	30.0	31.7	30.8
monos reco	88.6	100	8.5	13.1	10.4	4.0	58.6	7.5	4.8	7.6	5.9	1.7	1.1	1.3	30.6	32.6	31.6

Table 2: Results of the evaluation toolbox for the JHU, MODIS and AG systems (in percentage). Several outputs of JHU are evaluated separately, and the AG system is paired either with a gold transcription or an HMM phone recognizer.

which has only 59.2% phone accuracy. As our AG model is build under the assumption that word forms have a unique realization, it is not surprising to also witness a sharp drop in lexicon-related measures. Indeed, the variability of the phoneme transcriptions for each word triggers a multiplication of surface word forms that have to be segmented and stored as separate entries. Note that here, the more constrained monosyllable model gives comparable results to the unigram model. It would be interesting to improve the AG model by incorporating possibility to encode variations between the underlying forms and the surface forms. Possible directions for such extensions include modeling variability through noisy channel methods (Elsner et al., 2012), or replacing the sequence of phonemes as input by decoding lattices (Neubig et al., 2010).

We now turn to the evaluation of the ST systems at the phoneme level. The results on JHU and MODIS show a sharp decrement in coverage compared to the exhaustive coverage of the AG system: The matching stage of the JHU system has the highest coverage (36%), which then drops after clustering (28%), and drops again after parsing (24%). MODIS has the lowest coverage of all (8.4%), which is explained by a rather conservative DTW threshold compared to the JHU system. This conservative threshold in MODIS has some advantage, though, as it produces the highest matching F-score of the ST systems (12%, as opposed to less than 5% for the others). For the JHU system, the matching F-score is overall low, but the parsing component contributes to increase it. Regarding the clustering F-score, we can see that the JHU system is giving the highest grouping F-score when we take the output just after the clustering step, suggesting that its graph clustering method is quite effective (certainly in terms of recall). This advantage, however, is reduced after the parsing step. This is probably due to the fact that parsing eliminates a lot of nearly overlapping fragment candidates through the Viterbi decoding, and therefore many clusters are left as singletons. Maybe a different decoding step, favoring bigger clusters, would be more effective.

Before moving to the lexical level, it is interesting to note that some the best performing ST systems outperform the AG+reco pipelines on matching and grouping F-scores. This may seem surprising given that the ST systems work on raw speech, whereas the AG pipeline benefits from supervised training with an optimized HMM architecture. There are however two factors that may contribute to this result. The first one is that the AG pipeline does not exploit the full information of the phone recognizer (i.e. a decoding lattice), but only reads out the most probable parse. The second one is that due to a rather conservative matching threshold, plus a threshold on the minimal size of a matching fragment, the ST systems restrict their analyses on a small subset of the entire corpus, namely, a subset with good quality matches.

Finally, the lexical scores of the ST systems are very low, as expected. It is interesting, though, to compare them with the performance of the AG pipeline with phone recognition as input. In particular, there is a sharp difference in boundary F-score between the ST systems and the AG+reco systems, which seems mostly due to low recall in the ST systems. One reason AG systems are good are finding boundaries is that they impose an exhaustive match of the entire utterance, thereby using well identified frequent words to constrain the segmentation of less known surrounding materials. Such a strategy is typically not used in ST systems. It would be interesting to add such a strategy in the parsing scheme of the JHU system. Another reason is that AG, contrary to ST systems, does not have a minimal length on fragment matches. Both MODIS and JHU discard possible matches that are less than a fixed length in msec, because the clustering scores tend to decrease for short fragments because of the increased presence of lexical neighbors. Of course, a completely fair comparison would be to run the AG on some unsupervised phoneme decoder as in (Lee and Glass, 2012). This remains to be done in future work.

7. Conclusions

We have proposed in this paper a new evaluation method for TD systems. We consider that these systems have a huge potential in speech technology applications (Glass, 2012) and a common and complete evaluation procedure would help expand this young field. We have also aimed at bringing together the evaluation metrics of ST and NLP systems. While applications of these systems have already appeared ((Malioutov et al., 2007; Dredze et al., 2010; Muscariello et al., 2011) just to name a few), new, hybrid systems can be

developed, combining algorithms from both fields. These hybrid systems will benefit from our evaluation method which distinguishes the different components of the processing pipeline that link the signal to the linguistic levels. The proposed toolbox is available in github[3].

8. Acknowledgements

BL, MV, X-NC and ED's research was funded by the European Research Council (ERC-2011-AdG-295810 BOOTPHON), the Agence Nationale pour la Recherche (ANR-2010-BLAN-1901-1 BOOTLANG) and the Fondation de France. It was also supported by ANR-10-IDEX-0001-02 PSL and ANR-10-LABX-0087 IEC. MJ's research was partially supported under the Australian Research Council's *Discovery Projects* funding scheme (project numbers DP110102506 and DP110102593).

9. References

E. Amigo, J. Gonzalo, J. Artiles, and F. Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486.

N. Bernstein-Ratner. 1987. The phonology of parent-child speech. In K. Nelson and A. van Kleeck, editors, *Children's Language*, volume 6, pages 159–174. Erlbaum.

M. Brent and T. Cartwright. 1996. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61:93–125.

M. Brent. 1999. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34:71–105.

L. Catanese, N. Souviraà-Labastie, B. Qu, S. Campion, G. Gravier, E. Vincent, and F. Bimbot. 2013. MODIS: an audio motif discovery software. In *Proc. of INTERSPEECH 2013*, pages 2675–2677.

R. Daland and J. Pierrehumbert. 2011. Learning diphone based segmentation. *Cognitive Science*, 35(1):119–155.

R. Daland and K. Zuraw. 2013. Does korean defeat phonotactic word segmentation? In *Proceedings of the 51th Annual Meeting of the ACL*, pages 873–877.

M. Dredze, A. Jansen, G. Coppersmith, and K. Church. 2010. NLP on spoken documents without ASR. In *Proc. of EMNLP 2010*, pages 460–470.

M. Elsner, S. Goldwater, and J. Eisenstein. 2012. Bootstrapping a unified model of lexical and phonetic acquisition. In *Proceedings of the 50th Annual Meeting of the ACL*, pages 184–193.

R. Flamary, X. Anguera, and N. Oliver. 2011. Spoken WordCloud: Clustering recurrent patterns in speech. In *Proc. of Int. Workshop on Content-Based Multimedia Index*, pages 133–138.

A. Fourtassi, B. Börschinger, M. Johnson, and E. Dupoux. 2013. Whyisenglishsoeasytosegment? In *Proc. of CMCL 2013*.

J. Glass. 2012. Towards unsupervised speech processing. In *Proceedings of International Conference on Information Science, Signal Processing and their Applications 2012*, pages 1–4.

[3] github.com/bootphon

S. Goldwater, T. Griffiths, and M. Johnson. 2009. A bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54.

C. Herley. 2006. ARGOS: Automatically extracting repeating objects from multimedia streams. *IEEE Transactions on Multimedia*, 8(1):115–129.

A. Jansen and B. Van Durme. 2011. Efficient spoken term discovery using randomized algorithms. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pages 401–406.

A. Jansen, E. Dupoux, S. Goldwater, M. Johnson, S. Khudanpur, K. Church, N. Feldman, H. Hermansky, F. Metze, R. Rose, M. Seltzer, P. Clark, I. McGraw, B. Varadarajan, E. Bennett, B. Borschinger, J. Chiu, E. Dunbar, A. Fourtassi, D. Harwath, C. Lee, K. Levin, A. Norouzian, V. Peddinti, R. Richardson, T. Schatz, and S. Thomas. 2013. A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition. In *Proceedings of ICASSP 2013*, pages 8111–8115.

M. Johnson and S. Goldwater. 2009. Improving nonparameteric Bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings HLT - NAACL 2009*, pages 317–325.

M. Johnson, T. Griffiths, and S. Goldwater. 2007. Adaptor Grammars: A framework for specifying compositional nonparametric Bayesian models. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 641–648. MIT Press.

C. Lee and J. Glass. 2012. A nonparametric bayesian approach to acoustic model discovery. In *Proceedings of the 50th Annual Meeting of the ACL*, pages 40–49.

I. Malioutov, A. Park, R. Barzilay, and J. Glass. 2007. Making sense of sound: Unsupervised topic segmentation over acoustic input. In *Proc. of ACL 2007*, pages 504–511.

F. McInnes and S. Goldwater. 2011. Unsupervised extraction of recurring words from infant-directed speech. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*.

A. Muscariello, G. Gravier, and F. Bimbot. 2011. Zero-resource audio-only spoken term detection based on a combination of template matching techniques. In *Proc. of INTERSPEECH 2011*, pages 921–924.

A. Muscariello, G. Gravier, and F. Bimbot. 2012. Unsupervised motif acquisition in speech via seeded discovery and template matching combination. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(7):2031–2044.

G. Neubig, M. Mimura, S. Mori, and T. Kawahara. 2010. Learning a language model from continuous speech. In *Proc. of INTERSPEECH-2010*, pages 1053–1056.

A. Park and R. Glass. 2008. Unsupervised pattern discovery in speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):186–197.

M. Pitt, L. Dilley, K. Johnson, S. Kiesling, W. Raymond, E. Hume, and E. Fosler-Lussier. 2007. Buckeye corpus of conversational speech (2nd release).

[www.buckeyecorpus.osu.edu]. Columbus, OH: Department of Psychology, Ohio State University (Distributor).

M. Tekieli and W. Cullinan. 1979. The perception of temporally segmented vowels and consonant-vowel syllables. *Journal of Speech, Language and Hearing Research*, 22(1):103–121.