Computational Linguistics in the Netherlands Journal 3 (2013) 34-48 Submitted 05/2013; Published 12/2013

Automatic syllabification using segmental conditional random fields

Kseniya Rogova* KSENIYA.ROGOVA@ESAT.KULEUVEN.BE
Kris Demuynck** KRIS.DEMUYNCK@UGENT.BE
Dirk Van Compernolle* DIRK.VANCOMPERNOLLE@ESAT.KULEUVEN.BE

*KU Leuven, Kasteelpark Arenberg 10 bus 2441, 3001 Heverlee, Belgium

**Ghent University, Sint-Pietersnieuwstraat 41, 9000 Gent, Belgium

Abstract

In this paper we present a statistical approach for the automatic syllabification of phonetic word transcriptions. A syllable bigram language model forms the core of the system. Given the large number of syllables in non-syllabic languages, sparsity is the main issue, especially since the available syllabified corpora tend to be small. Traditional back-off mechanisms only give a partial solution to the sparsity problem. In this work we use a set of features for back-off purposes: on the one hand probabilities such as consonant cluster probabilities, and on the other hand a set of rules based on generic syllabification principles such as legality, sonority and maximal onset. For the combination of these highly correlated features with the baseline bigram feature we employ segmental conditional random fields (SCRFs) as statistical framework. The resulting method is very versatile and can be used for any amount of data of any language.

The method was tested on various datasets in English and Dutch with dictionary sizes varying between 1 and 60 thousand words. We obtained a 97.96% word accuracy for supervised syllabification and a 91.22% word accuracy for unsupervised syllabification for English. When including the top-2 generated syllabifications for a small fraction of the words, virtual perfect syllabification is obtained in supervised mode.

1. Introduction

The definition of the concept "syllable" depends on the dictionary or author one consults. The American Heritage dictionary (2011) defines syllable as "a unit of spoken language consisting of a single uninterrupted sound formed by a vowel, diphthong, or syllabic consonant alone, or by any of these sounds preceded, followed, or surrounded by one or more consonants". Syllables are often considered the phonological "building blocks" of words. They influence the rhythm of a language, its prosody, its poetic meter, its stress patterns, etc. (Blainey 1993).

A syllable is composed of an onset (set of consonants), a nucleus (vowel) and a coda (set of consonants) (Huang et al. 2001). Languages differ with respect to various typological parameters, such as optionality of onsets and the allowed complexity of the syllable constituents. Given the task of counting the number of syllables in an utterance, naive listeners will have little difficulty, though they have great difficulties when asked to point where exactly the syllable boundaries lie (Goslin and Frauenfelder 2001). Some languages, like Chinese, have strict syllabification rules while even linguists argue about the correct syllabification in other languages, like English (Duanmu 2009). This inherent ambiguity in syllabification combined with further ambiguity in phonetic transcriptions leads to a poor 84% consistency between English syllabifications in two widely used corpora, namely CELEX and Merriam-Webster Online (Bartlett et al. 2009).

If the syllable does act as a structuring device in composing words, as many believe, then knowing this structure could aid word modeling in automatic speech recognition and/or the unit selection and composition process of concatenative synthesis (Marchand et al. 2009). The syllable is an attractive unit for several reasons: it is much more salient than the shorter phone-unit, co-articulation tends to be stronger within than across syllables and it is the shortest unit that contains all acoustic attributes relating to phonetics, rhythm and prosody. Hence, the pronunciation of a phoneme may depend on its exact location within a syllable.

So, arguments about the theoretical status of the syllable as a linguistic unit notwithstanding, there are good practical reasons for seeking powerful algorithms to syllabify words (Marchand et al. 2009).

While syllables should be appealing for speech technology, they are not as widely used as one could expect. One of the reasons is that syllabified resources are still limited and of poor quality in many languages. This is due to the inherent ambiguities of the syllabification process, the fact that proper names and loan words follow different rules than the core of a language and the intrinsic poor quality of certain resources.

The main aim of this research is to make a universal syllabification tool that is language-independent and can operate with various amounts of syllabified training data. The syllabification tool we developed in this work addresses the above mentioned problems: it assures complete coverage and learns equally well from small or large example corpora. The system takes as input the phonetic transcription of a word and adds the syllable boundaries. Hence our tool forms a nice complement for languages with ample phonetic resources though with limited syllabic resources.

Our approach combines a probabilistic formulation of the legality, sonority and maximal onset principles with co-occurrence statistics in a single model. The only language-dependent a priori information needed is the set of phones and the sonority scale. Other information is derived from a large corpus of data (with or without syllabification information). The approach is suitable for different languages and trainable in supervised, unsupervised and semi-supervised mode. Individual knowledge sources such as the co-occurrence statistics are estimated in a maximum likelihood way and the scores derived from these correlated indications are combined using segmental conditional random fields. The model returns all possible syllabifications and corresponding probabilities in line with the inclusion of 'pronunciation' variants in phonetic dictionaries. In practice only one and occasionally two variants are retained.

The remainder of the paper is organized as follows. We give an overview of background material and related work in Section 2. Section 3 discusses the syllable bigram model. The remaining syllable features are described in Section 4. The segmental conditional random fields framework and its application for syllabification are explained in Section 5. Training strategies and multiple output are presented in Section 6. Section 7 presents the experiments and results. Section 8 gives an interpretation to the results and Section 9 concludes the paper.

2. Background and related work

Syllable structure

First, we need to formally define a syllable. In this work, a syllable (S) is a sequence of phones, consisting of three parts: an onset (O), a nucleus (V) and a coda (C). The nucleus consists of a single vowel or diphthong and is the obligatory part of a syllable. Onset and coda are composed of a sequence of consonants and are optional. Each word may be written as a sequence of syllables. To indicate the word beginning and word ending we use $\langle \mathsf{w} \rangle$ and $\langle / \mathsf{w} \rangle$ symbols.

$$\begin{aligned} word &\longrightarrow \langle \mathsf{w} \rangle\, S_1\, S_2 \cdots S_N\, \langle / \mathsf{w} \rangle \\ S_i &\longrightarrow O_i V_i C_i \end{aligned} \tag{1}$$

Related work

There are two main ways of automatic syllabifications: rule-based and data-driven (Marchand et al. 2007). The rule-based approach is based on generic principles for syllabification. Data-driven methods use a syllabified corpus to make a syllabification of the unknown word. Marchand and his colleagues (2007, 2009, 2009) showed that rule-based systems perform poorly compared to the data-driven methods.

SYLLABIFICATION BASED ON UNIVERSAL PRINCIPLES

There are three well established and generic principles for syllabification, namely the maximal onset principle (Kahn 1976), the sonority sequencing principle (Selkirk 1984) and the legality principle (Goslin 2002).

The sonority principle assigns numerical values to every phone of a syllable along the sonority scale where vowels have the highest rank followed by nasals, fricatives and plosives. Ranks of consecutive phones increase in the onset and decrease in the coda, apart from some language-dependent exceptions. For example, the English onsets `sp,sk,str` do not fit the sonority rule though they are valid. The main problem of this principle is the inability of obtaining the correct syllabification when several splits are valid. The main merit of the method is its independence of the training data set.

The legality principle allows consonant clusters to be valid initial or final only if they appear as initial or final clusters of syllables/words. It means that a syllable onset is legal only if such onset was seen as initial phone sequence in one of the words from the training data. The same idea is used for syllable codas. To apply the legality principle, we need a big corpus though not necessarily syllabified. Legality has the same main drawback as sonority: when several valid splits of the consonant cluster between the vowels are possible, the syllabification is ambiguous.

The maximal onset principle gives preference to longer onsets if multiple legal splits of a consonant cluster are possible. This principle obtains a unique classification, though training data is still essential to obtain a list of valid onsets and codas.

SYLLABIFICATION USING DATA DRIVEN APPROACHES

Syllabifying purely based on the syllabification principles is sometimes inadequate as in practice the rules may be insufficient to disambiguate in some situations. More than that, there are cases when correct syllabification breaks one of the principles. Hence a more fine grained interpretation, typically some statistical formulation of these principles and the combination thereof, has given rise to a number of data driven syllabification methods that mainly differ in how these principles are incorporated in a model and how the model parameters are estimated from a corpus of example data.

Muller (2006) developed grammars to describe the phonological structure of words. To increase the prediction precision of syllable boundaries, she introduces fine-grained grammars to better learn the phonotactic information. Using grammars, a word is presented as a syllable sequence. Each syllable splits into an onset and a rhyme. The rhyme at the same time is written as a nucleus and a coda. Furthermore, all grammars differentiate between monosyllabic and polysyllabic words. Additionally, the grammars distinguish between consonant clusters of different size.

Another approach was suggested by Zhang and Hamilton (1997). They presented the LE-SR (Learning English Syllabification Rules) system, which learns rules using a symbolic pattern recognition approach. Each grapheme in a word is translated into C-S-CL representation. “C” stays for a consonant; “S” - for a syllabic grapheme and “CL” - for a consonant cluster. Syllabification rules and cutting patterns are learned though a syllabified corpus. To determine which cut should be chosen as a candidate rule, the authors combine a statistical approach with a symbolic pattern recognition approach and calculate the frequency of each cut.

Ananthakrishnan (2004) looked at the syllabification problem as searching for the most probable syllable bracketing given a phoneme sequence. The author used a statistical approach with supervised and unsupervised learning. He simplified the probability of a syllabification given the nuclei to the product of probabilities seeing the onset and coda given the previous, current and following nucleus. This method bears some resemblance to ours; however, it employs a different model parametrization and parameter estimation approach.

Bartlett suggested a discriminative approach that combines Support Vector Machine and Hidden Markov Model technologies and achieved one of the best published results (Bartlett et al. 2009). A multiclass SVM was used to classify each phoneme according to its position in a syllable on the basis of a set of features. The HMM overcomes the problem of treating each phoneme in a word independently. When training a structured SVM, each training instance (word) is paired to its label (syllabification as sequence of onset/nucleus/coda), drawn from the set of possible labels. The SVM finds the best separator between correct and incorrect tagging.

Ouellet and Dumouchel described a heuristic syllabification method that works by assigning costs to consonant clusters and then splitting the clusters where the cost is minimized (Ouellet and Dumouchel 2001).

The authors distinguish "good" and "bad" onsets and codas collectively called consonant cluster (CC). A CC is bad until proven good. Good onsets and codas are the ones found frequently in the training data.

The syllabification by analogy (SbA) approach works in a similar way as pronunciation by analogy (Marchand et al. 2007). When a word with unknown syllabification is presented as an input to the system, so-called full pattern matching between the input string and database entries is performed.The decision function identifies the "best" candidate according to some criterion.

Hammond (1997) applied Optimality Theory to syllabification that is treated in terms of constraint hierarchies. The basic idea is that every possible syllabification of an unsyllabified input string is generated and then evaluated with respect to the constraint set. Extremely important aspects of this system are that the constraints are violable and strictly ranked.

Connectionist networks incorporate a morphological parser and lexicon (Daelemans and van den Bosch 1992). The work is based on maximal onset and sonority principles and a window encoding approach. The decision whether the phone is the first character of a new syllable is made given a certain character position in a word and both left and right contexts (using from one to three phones on each side).

A statistical method for the segmentation of words into syllables based on a joint n-gram model was explored by Schmid et al. (2007). They chose a tagging approach i.e. the program annotates each phone symbol in the transcription of a word either with a "B" tag (indicating a syllable boundary after the phone) or an "N" tag (no syllable boundary). The syllable tagger uses the Viterbi algorithm to efficiently compute the most probable tag sequence.

Another simple statistical approach was suggested by Mayer (2010). He counted the actual syllables in order to determine the best split of word-medial consonant sequences.

Goldwater and Johnson presented a language-universal rule-based algorithm that finds a good set of parameters and then trains them using EM (Goldwater and Johnson 2005). Their model is defined by a grammar which describes the syllable structure. Their method differs from other similar approaches by learning the parameters in an unsupervised manner. Two models were investigated: positional and bigram.

There are a number of research works on comparison of different syllabification methods. Marchand et al. (2009) and Pearson et al. (2000) proved that data driven methods (decision tree and global statistics) are more effective than rule-based approaches.

3. Syllabification model

The formulation in (1) implies a number of syllables equal to the number of vowels (and/or diphthongs) and reduces the syllabification problem to finding correct segmentations in the consonant clusters occurring in between vowels. We solve the above problem by finding the most likely sequence of syllables that explains the phonetic transcription of the word.

The probability of a syllabification for a given word $P(syl|word)$, can be computed by applying the probability chain rule:

$$P(syl|word) = P(S_1 \mid \langle \mathsf{w} \rangle) \times P(S_2 \mid S_1, \langle \mathsf{w} \rangle) \times \ldots \times P(\langle / \mathsf{w} \rangle \mid S_N, \cdots, S_1, \langle \mathsf{w} \rangle)$$

This is simplified by applying a bigram constraint on the syllable sequence model:

$$P(S_i|S_{i-1}, .., S_1, \langle \mathsf{w} \rangle) \approx P(S_i|S_{i-1}) \tag{2}$$

The bigram approximation seems very reasonable as its operating window is wide enough to incorporate all general syllabification principles and furthermore allows for the incorporation of fine grained language specific knowledge in terms of syllable existence and syllable sequence information at the bigram level.

Syllable bigram model

The straightforward way to calculate the probability $P(S_i|S_{i-1})$ is the maximum likelihood estimate based on simple counting. This maximum likelihood estimate is supplemented with Good-Turing discounting and Katz back-off (Huang et al. 2001) to smooth those estimates most affected by sparsity.

In practice, we derive three logarithmic scores, namely a bigram score $Q_b(S_{i-1}S_i) = \log P(S_i|S_{i-1})$, a unigram score $Q_u(S_i)$, and a unigram back-off score $Q_{ubo}(S_{i-1}S_i)$ These LM parameters are calculated using the SRILM toolkit (Stolcke 2002). Scores for unseen bigrams are computed from the unigram and unigram back-off scores $Q_b(S_{i-1}S_i) = Q_{ubo}(S_{i-1})$ + $Q_u(S_i)$. In our implementation the score for unseen syllables $Q_u(\langle \mathrm{unk} \rangle)$ is approximated as the score for syllables seen once. Note that $Q_{ubo}(\langle \mathrm{unk} \rangle) = 0$.

Experiments

The bigram language model was evaluated on the CELEX database (Baayen et al. 1996). We randomly selected 5K words for testing. We obtained a fair 93.42% word accuracy (percentage of correctly syllabified words) when using 50K words as development data. When using only 5K words development data, the word accuracy drops significantly to only 76.3%.

The main problem for the syllable bigram approach is clearly data sparsity. Figure 1 shows out-of-vocabulary (OOV) rates for syllables, some of its constituents and certain combinations. Consonant clusters (i.e. consonants between two vowels) marked with internal syllable boundaries are denoted by `CCSplit`. `CC` stands for consonant clusters without syllable boundary information. To compute the statistics we created multiple train and test sets from the CELEX dictionary. The size of the training set was varied; the size of the test set was fixed to 5K words.

The syllabification results obtained in the above experiment can easily be explained by looking at the OOV rates of the syllable bigrams. Around 40% of all syllable bigrams in the 5K test words were not seen in the 50K words development data. This value rises to 78% when decreasing the development set to 5K words. Even single syllables have a quite high OOV rate. These values vary from 13% to 46% when using different amounts of development set. The back-off mechanisms applied in n-gram language modeling and applied in the above setup are neither intended nor optimal for such high OOV-rates. Such situations call for a back-off to principles or features that are simple enough to be handcrafted (rule-based) or trainable from very small amounts of data. We therefore introduce such additional features in the next section.

4. Features

For large training corpora the bigram model works relatively well as it embeds in its parameters the general syllabification principles and more. However, it exhibits considerable degradation with decreasing database size. For these situations and in general as OOV remains considerable, it was found beneficial to add the general principles (sonority, legality and maximum onset) and some robust statistical scores, which may be considered a means of class based back-off. All features have a monotonous log-prob like "goodness of fit" characteristic and are scaled to be approximately in the range 0 to -1. A score of 0 implies a perfect fit between feature and hypothesis , while -1 shows clear counter evidence. For feature definitions we use O_i for onset, C_i for coda and CC for consonant cluster.

The sonority feature is calculated for onset and coda separately. We define the sonority scale with nine different levels starting with voiceless stop and finishing with vowels. A level is assigned to each phone. An onset fits the sonority principle if the sonority level of the phones are increasing. Similar, a coda fits the principle if the corresponding levels decrease.

$$f_s(cc) = \begin{cases} 0 & \text{if } cc \ (O_i \ or \ C_i) \text{ fits the sonority principle} \\ -1 & \text{otherwise} \end{cases} \tag{3}$$

Examples:
onset `pl`: $f_s(pl) = 0$; coda `nd`: $f_s(nd) = 0$; onset `kt`: $f_s(kt) = -1$; coda `bj`: $f_s(bj) = -1$

The legality feature is also computed for onset and coda individually. We call an onset (coda) legal, if it can be found in the list of known onsets (codas). The list of valid onsets and codas is collected from the

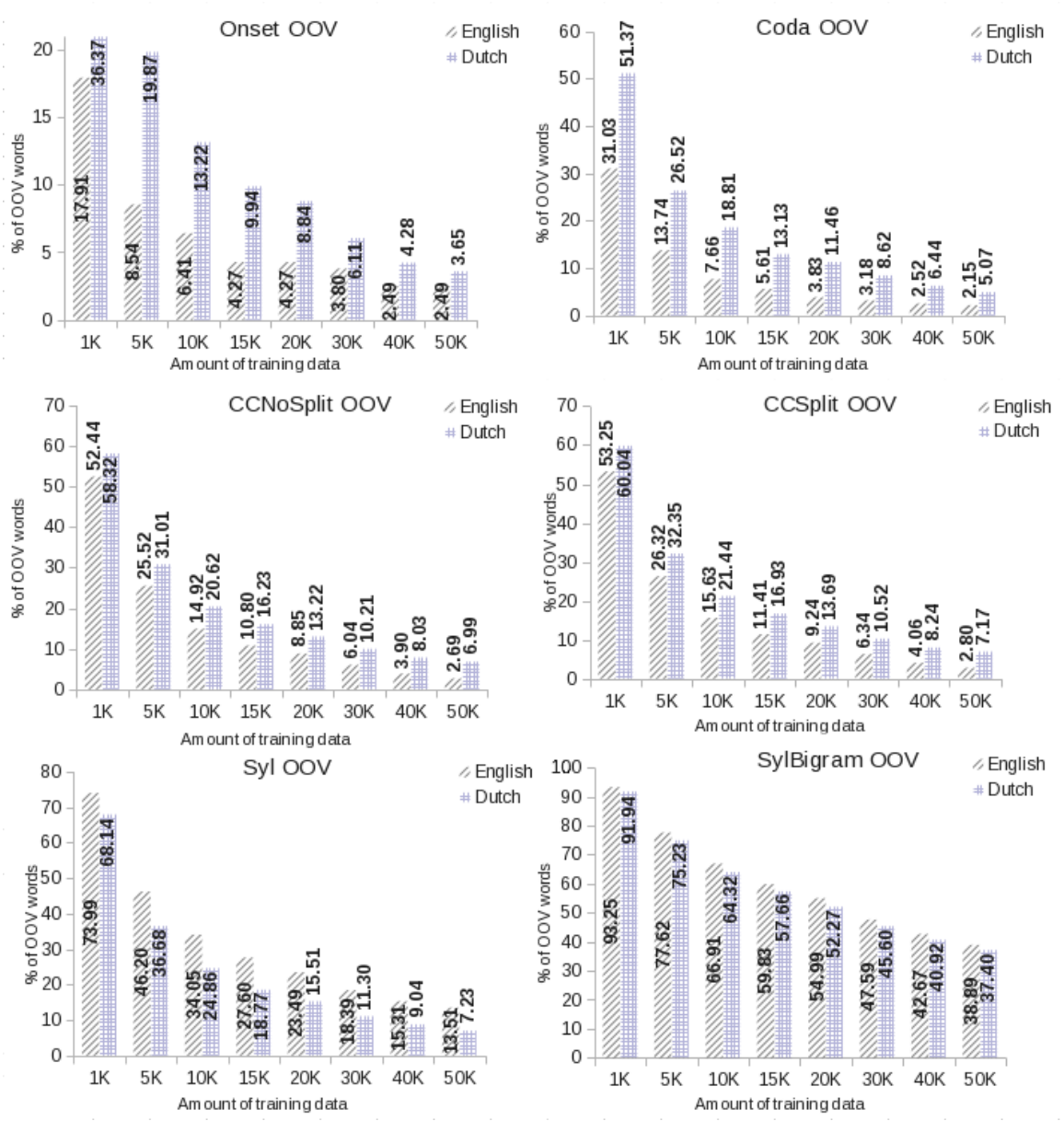

Figure 1: OOV rate for different amounts of development data, measured on 5K testing data for CELEX English and Dutch.

development data. We include both onsets and codas of all syllabified words as well as word initials and finals of non-syllabified words.

$$f_l(cc) = \begin{cases} 0 & \text{if } cc \text{ } (O_i \text{ } or \text{ } C_i) \text{ is legal} \\ -1 & \text{otherwise} \end{cases} \quad (4)$$

Example:
onset `st`: $f_s(st) = 0$; coda `ks`: $f_s(ks) = 0$; onset `ns`: $f_s(ns) = -1$; coda `ln`: $f_s(ln) = -1$

The MaxOnset feature is computed for every consonant cluster between two vowels. In our implementation the feature is slightly modified: onset and coda legality are not checked. These cases are handled by the legality feature.

$$f_o(C_{i-1}O_i) = \begin{cases} 0 & \text{if } \mathrm{len}(C_{i-1}O_i) = 0 \\ \frac{\mathrm{len}(O_i)}{\mathrm{len}(C_{i-1}O_i)} - 1 & \text{otherwise} \end{cases} \tag{5}$$

Example: consonant cluster `kstb`:
$f_o(\epsilon\_kstb) = 0$ $f_o(k\_stb) = -0.25$ $f_o(ks\_tb) = -0.5$ $f_o(kst\_b) = -0.75$ $f_o(kstb\_\epsilon) = -1$

The CC feature reflects the consonant cluster split. This feature indicates the probability of a syllable boundary in a specific place of the consonant cluster. The probability is calculated using occurrence statistics derived from the development corpus. For a consonant cluster that never occurred in the development data, the probability is approximated as 1 over the number of consonant clusters in the development corpus. Notation ct stands for count.

$$f_{cc}(C_{i-1}, O_i) = \log P(O_i \mid C_{i-1}) = \begin{cases} \log \dfrac{ct(C_{i-1}O_i)}{ct(C_{i-1})} & \text{if } ct(C_{i-1}) \neq 0 \\ \log \dfrac{1}{ct(CC)} & \text{otherwise} \end{cases} \tag{6}$$

Example:
$f_{cc}(k, s) = -0.908$ $f_{cc}(\epsilon, kw) = -0.054$ $f_{cc}(k, stb) = -1.792$

The syllable probability feature indicates the probability of the syllable given the onset. It is calculated using occurrence statistics derived from the development corpus. The notation $O_i + *$ stands for syllables with onset O_i; $ct(syl)$ is the total number of syllables in the corpus. When there are no syllables starting with a specific onset, the probability is approximated as 1 over the number of different syllables in the corpus.

$$f_{syl}(S_i, O_i) = \log P(S_i \mid O_i) = \begin{cases} \log \dfrac{ct(S_i)}{ct(O_i + *)} & \text{if } ct(O_i + *) \neq 0 \\ \log \dfrac{1}{ct(syl)} & \text{otherwise} \end{cases} \tag{7}$$

Example:
$f_{syl}(ftri, ftr) = -16.118$ $f_{syl}(tIk, t) = -4.352$

5. SCRF framework

The conditional random fields (CRF) framework models the conditional probability $P(y \mid x)$ of a label sequence y given the input sequence x. A CRF on (x, y) is specified by a vector f of local features and corresponding weight vector λ. Each local feature is either a state or a transition feature. The global feature vector for input sequence x and label sequence y is given by $F(y, x) = \sum_i f(y, x, i)$, where i ranges over input positions (Sha and Pereira 2003). The label sequence can be found through the Viterbi algorithm.

Then conditional probability is defined by

$$p_\lambda(y \mid x) = \frac{\exp \lambda \cdot F(y, x)}{Z_\lambda(x)}$$

where

$$Z_\lambda(x) = \sum_y \exp \lambda \cdot F(y, x)$$

Segmental CRF (SCRF) extends CRF model by operating at the segment level, in which multiple adjacent observations can be lumped together into a segment with a single label, and segment-level features can be extracted and used (Zweig and Nguyen 2010).

SCRF for syllabification

In this work we apply the SCRF framework for syllabification by modeling the conditional probability of a syllable sequence (labels) given the phonetic transcription of the word (the observations) as a log-linear combination of feature functions. In the SCRF framework the bigram model is just one of a variety of features. Advantages of SCRF are that they can deal with many statistically correlated features, control overfitting and have a good mechanism for parameter estimation. Hence it is very versatile and can be used for any amount of data of any language. We operate SCRF on the syllable level; therefore all the features are also computed for syllables.

Let N be a number of syllables (number of vowels or diphtongs) and M a number of features. For uniform presentation the syllable bigram probability described above (Section 3) is called the syllable bigram feature $f_j(S_i \mid S_{i-1}) = \log P(S_i|S_{i-1})$.

$$\begin{aligned} Score(S_i|S_{i-1}) &= \sum_{j=1}^{M} \lambda_j \cdot f_j(S_i \mid S_{i-1}) \\ Score(syl|word) &= \sum_{i=1}^{N+1} Score(S_i|S_{i-1}) \\ S_0 = \langle \mathsf{w} \rangle; &\quad S_{N+1} = \langle / \mathsf{w} \rangle \\ P(syl|word) &= \exp Score(syl|word) \end{aligned} \tag{8}$$

In our work $M = 8$: 2 sonority features (for onset and coda), 2 legality features (onset and coda), maximal onset, consonant cluster, syllable and syllable bigram features.

6. Training strategies

There are two levels of training: (1) training of the statistical features and (2) training of the weights in the SCRF. For this purpose the total training data is divided into a development set used for feature training and a validation set used for weight training. The scheme is easily adaptable to supervised, unsupervised and semi-supervised training strategies. In the case of supervised training the feature training is done in a single maximum likelihood pass, followed by gradient descent weight training in the SCRF. In the case of unsupervised and semi-supervised we iterate over both feature and weight training as segmentations of the train corpus improve throughout the process. A bootstrap phase derives initial estimates for all features from monosyllabic words, onsets and codas, and the partially segmented corpus. In unsupervised mode the global training is re-split between development and validation sets after each iteration. In semi-supervised mode a preset fraction of the training data is set apart as validation set. The general scheme is shown in Figure 2.

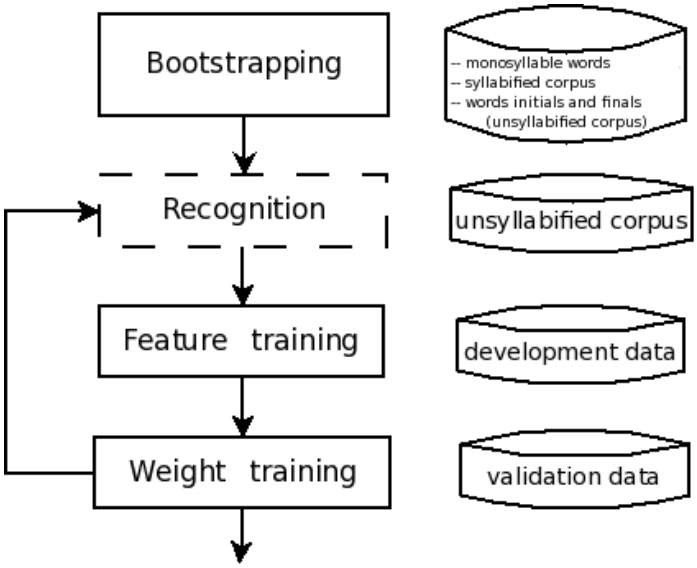

Figure 2: Syllabification algorithm

Multiple output

Syllabification is not uniquely defined and is to some extent influenced by the word morphology. The ambiguity in the syllabification process and the underlying phonemic transcriptions leads to a substantial inconsistency between human transcribers. Bartlett (2009) for example reported that only 84% of words have identical syllabification when comparing CELEX (Baayen et al. 1996) with Merriam-Webster Online.

The first cause of ambiguity is disagreement in the phonetic transcription of words. Using IPA symbols, CELEX transcribes "serious" and "category" as sɪərɪəs and kætəgərɪ respectively whereas Wordsmyth (2011) transcribes them as siːriːəs and kætəgəʊriː.

The second problem, the ambiguity inherent to the syllabification process, can be observed in inflections of the same lemma within the same database and with consistent phonetic transcriptions. This also includes simple mistakes and typos that can be in any dictionary. For example, in CELEX we found the word discipline syllabified as `dI-sI-plIn` though the word indiscipline was syllabified like `In-dI-sIp-lIn`.

7. Experiments and results

7.1 Experimental setup

Our experiments were carried out on the CELEX database and Wordsmyth dictionary. We randomly divided our databases into development, validation and testing sets. The size of the development set is varied from 50K words to 1K words. The size of the validation set is chosen proportional (20%) to the size of the development set and ranged from 10K to 200 words. The size of the test set is fixed to 5K words. To average out the effect of random splits, each experiment was repeated 5 times. We used English and Dutch languages to carry experiments.

To assess the different methods, we report *word accuracy* (percentage of correctly syllabified words) and *syllable accuracy* (percentage of correctly detected syllables). We also report on the case in which multiple syllabifications are retained. In practice either one or two syllabifications are considered. The decision to return a second syllabification depends on the rank ordering of the probability ratios between the

top-2 choices. The threshold was chosen so that either 0%, 10% or 100%[1] of the words receive a second syllabification variant.

We used the SCARF toolkit developed by Microsoft (Zweig and Nguyen 2010) to estimate the parameters in the SCRF (Equation 8). Although the SCARF toolkit is originally designed for speech recognition tasks, it can be readily adopted for other tasks such as syllabification.

7.2 Results

We first conduct a series of experiments based on supervised training. Figure 3 shows word accuracy in function of development set size. The solid curve shows the results for the single best output and the dashed curves for 10% and 100% of the words with 2 syllabification variants.

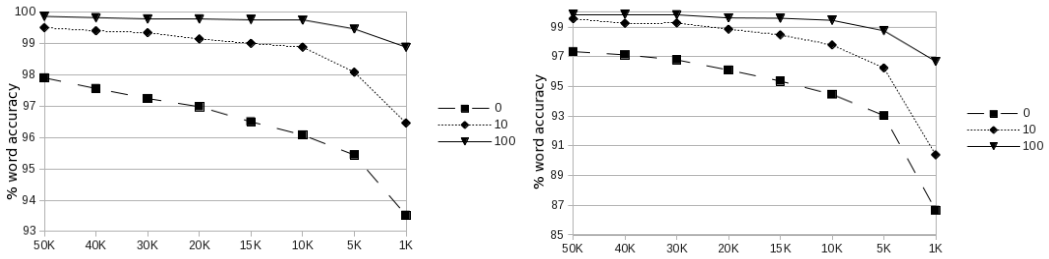

Figure 3: Word accuracy in function of the amount of development data for English (left) and Dutch (right).

Table 1 lists the results for supervised training with various amounts of syllabified data (from 50K to 1K words) for both English and Dutch. We achieve 97.96% word accuracy for English and 97.35% for Dutch with the 50K development data.

	development data	50K	40K	30K	20K	15K	10K	5K	1K
	validation data	10K	8K	6K	4K	3K	2K	1K	200
English	word accuracy	97.96	97.64	97.47	97.07	96.71	96.49	95.67	93.59
	syllable accuracy	98.42	98.17	97.93	97.73	97.37	97.05	96.56	95.08
Dutch	word accuracy	97.35	97.13	96.79	96.11	95.37	94.47	93.04	86.67
	syllable accuracy	98.49	98.37	98.18	97.77	97.34	96.82	95.96	92.19

Table 1: Syllabification accuracy (% correct words/syllables) for supervised training

Similar experiments were performed on the English Wordsmyth dictionary. Taking 30 thousands words of development data, a 95.79% word accuracy and a 96.76% syllable accuracy was achieved.

These results are slightly worse (about 2.5% absolute) than for CELEX with the same amount of development data. We attribute these differences to a different structure of the dictionary. CELEX has a higher ratio of word forms versus lemmas which results in significantly better coverage of syllable bigrams and lower OOV rates for the longer units.

In a second series of experiments, we evaluate the performance of unsupervised and semi-supervised training. We work with a 60K data set. Unsupervised training was performed on 60K unsyllabified words. The semi-supervised training combined 6K words of syllabified data with 54K unsyllabified words. A syllabified corpus can be split in different ways between development and validation sets. We use three splits: 5K, 3K and 0K words for development data and corresponding 1K, 3K and 6K words for validation sets. Results

1. With '100%' we indicate that a second variant is included for all multi-syllabic words. The actual percentage depends on the fraction of multi-syllabic words in the test set.

are presented in the Table 2. The results show that it is advantageous to use most of the syllabified data for feature training. Weight training is done reliably with a very small amount of held out data.

unsyllabified words	syllabified words development	validation	English	Dutch
54K	5K	1K	96.14	94.18
54K	3K	3K	95.54	93.44
54K	0K	6K	93.42	86.24
60K	0K		91.22	85.49

Table 2: Word syllabification accuracy for unsupervised and semi-supervised training

None of the variations give 100% syllabification accuracy. Most mistakes occur in compound words such as `elsewhere`, `strongarm`, `placekicks`. Another source of errors are prefixes like `dis`, `on`, `in`, `out`. These confusions appear because correct syllabifications coincide with word boundaries. These boundaries may be overruled by general syllabification principles. The other errors occur from the statistics aspect of the development corpus. Some examples are presented in the Table 3. The mentioned sources of errors (compounding and suffixes) also explain the higher error rates in Dutch vs. English as Dutch is morphologically more productive than English.

word	correct syllabification	obtained syllabification
riskiness	rɪsk-ɪ-nɪs	rɪs-kɪ-nɪs
samplers	sɑːm-pləz	sɑːmp-ləz
wizardry	wɪ-zə-drɪ	wɪ-zəd-rɪ
walkouts	wɔːk-aʊts	wɔː-kaʊts
blondest	blɒn-dɪst	blɒnd-ɪst

Table 3: Examples of syllabification errors

7.3 Model evaluation

All results presented thus far are based on a SCRF that combines all features. In a final set of experiments we evaluated the contribution of the respective components. Table 4 gives the results for CELEX English supervised training with 50K words development set and 10K words validation set. These experiments show that each feature has its merit.

models included	50K	40K	30K	20K	15K	10K	5K	1K
bg feature	93.42	92.60	90.98	89.00	86.48	83.06	76.30	56.88
bg feature + son, leg, max onset	97.32	97.02	96.56	96.50	95.90	95.52	95.22	93.02
bg feature + cc, syl prob	97.6	97.14	96.64	96.00	95.34	94.06	92.02	81.80
all features	97.96	97.64	97.47	97.07	96.71	96.49	95.67	93.59

Table 4: Word syllabification accuracy for supervised training for English using selected model components

The bigram model works relatively well for large syllabified training corpora but the result drops significantly when the training set becomes small. With 1K training data, we reach only 56.88% word accuracy. Combining the sonority, legality and maximal onset features with the bigram model achieves higher results than the bigram model combined with consonant cluster and syllable features for small amounts of data. The opposite tendency is seen for large amounts of labelled training data. These dependencies have logical explanations. This is a direct correlation of the data sparsity problem illustrated before in Figure 1: although the consonant cluster and syllable statistics can model the phonotactics with more detail than generic principles such as sonority, legality and maximal onset, such detailed statistics require large corpora to be estimated reliably.

7.4 Feature evaluation

The feature weights λ_i assigned to the different features in a SCRF give some indication as to how important each feature is. When performing syllabification and feature weight estimation, it is interesting to compare the feature weights. The weights are listed in Table 5. Although the absolute values have little meaning (the scale of each feature differs), one can look at the relative change of the weights in function of the amount of training data.

feature	50K	40K	30K	20K	15K	10K	5K	1K
bigram	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
sonority(onset)	0.20	0.17	0.24	0.26	0.31	0.31	0.55	1.85
sonority (coda)	0.38	0.47	0.55	0.61	0.68	0.90	0.72	7.42
legality (onset)	1.37	1.18	1.83	1.26	1.62	2.03	1.88	5.77
legality (coda)	1.37	1.18	1.83	1.26	1.62	2.03	1.88	5.77
maximal onset	0.84	0.91	0.94	0.99	1.05	1.47	1.58	3.73
consonant cluster	1.02	1.11	1.11	1.09	1.31	1.53	3.03	2.65
syllable	0.90	0.82	0.96	1.10	1.37	1.41	1.51	5.48

Table 5: Feature weights for supervised training

There are three features based on co-occurence statistics, namely bigram model score, consonant cluster score and syllable score. With a bigger syllabified corpus these scores become more reliable and hence are more informative representing the general dependence between syllables and syllable parts. Although maximal onset and legality principles are also based on the training corpus they can be bootstrapped from unsyllabified data using word initials and finals. The overall effect is that when more syllabified data is used, the importance of the co-occurence statistics features (language model, consonant cluster and syllable) grows.

8. Discussion

The results presented in Table 4 prove that a system based on corpus based statistics can be complemented with a statistical formulation of universal syllabification principles. The features based on corpus statistics such as syllable bigram model, consonant cluster and syllable scores remain the most important features. However, the extra features based on generic syllabification principles create a far more robust system that is applicable across languages and that scales smoothly from unsupervised training to various degrees of supervision. Segmental conditional random fields provide a statistical framework that makes the integration of such a wide set of statistically dependent features practical and the discriminative nature of the parameter estimation process in turn results in a more accurate system.

The results presented are among the best published results so far, although comparisons are difficult by lack of standardization for the considered task. Even within CELEX results are not strictly comparable by lack of standardized train and test sets. We, in any case, tried to minimize these effects by 5-fold cross-validation. The setup used by Anantharishan (2004) is very similar to our English CELEX experimental

setup. He reported a 91.4% syllable accuracy, which is considerably worse than the 97.96% we obtained. Other results on comparable data are the 92.62% obtained by Muller (2006) and the 95.52% syllable accuracy reported by Zhang (1997), which are also worse than ours. The best result for supervised syllabification was reported by Bartlett et al. (2009). They obtained 98.86% word accuracy when using 30K words for training. The use of 4 phones before and after the central phone in combination with the position dependent modeling makes that this model can learn the influence of morphology on syllabification. Our approach, despite being far less complex still achieves competitive results. On the other hand our method is superior when only small amounts of data (1K words) are available (71% versus 93.59%). Furthermore, using our method's capabilities to output multiple variants, we can easily obtain over 99% word accuracy using only 30K words for training by outputting 2 variants for 10% of the words.

For unsupervised syllabification, Goldwater and Johnson (2005) report an accuracy of 97.1% for English. Since they do training and testing on running text, a direct comparison of results is problematic. On the one hand there is a higher weight given to the error free mono-syllabic words and on the other hand all frequent multi-syllabic words are emphasized during training. Just taking into account the effect of word frequencies in the test set we achieve a 97.07% syllabification accuracy as well.

Finally, let's revisit the option to retain multiple variants. Inspecting such variants, lets us conclude that a significant fraction of these are genuine variants that may be present in real life speech. Hence, while some may be hiding unresolved errors, others should be considered as genuine alternatives that should be included in applications such as speech recognition. Accepting alternatives reflects the fact that a certain amount of ambiguity is to be expected given the ambiguity of certain syllabification rules and the substantial inconsistency between human transcribers. Given such problems, one should not expect the model to come up with a single solution.

9. Conclusions

In this paper, we presented a simple and robust statistical framework for language-independent probabilistic syllabification of phonetic transcriptions of words using segmental conditional random fields. The method was evaluated on two different datasets and on two different languages and with variable amounts of supervision during training. It is shown that the method generalizes well with limited amounts of data and is widely applicable. The results of this work will be used in modeling automatic speech recognition system.

The use of a compact set of features which are based on well established principles and robust statistics prevents the system from memorizing the exceptions in the training data (outliers or even plain errors). The resulting consistency of the word syllabification is expected to be beneficial for our envisaged applications, namely speech recognition and speech synthesis.

The system obtained a 97.96% word syllabification accuracy on English words. Taking the top two syllabification variants into account for 10% of the words, improves the word accuracy to 99%.

Despite the fact that the model relies on a straightforward and easy to evaluate probabilistic model, it performed as good as other state-of-the-art systems. Furthermore, the system surpasses existing systems in two ways: (1) our model can output and rank multiple variants, reaching virtual perfection when including from 5% to 10% words with multiple variants, and (2) the method works well for supervised, unsupervised and semi-supervised training.

Acknowledgments

We thank the team of the Wordsmyth for providing their data. We also thank G. Zweig for help with SCARF adaptation. This research was supported by the Fund for Scientific Research Flanders (Projects "TELEX" G.0260.07 and "AMODA" G.A122.10N).

References

Adsett, Connie R. and Yannick Marchand (2009), A comparison of data-driven automatic syllabification methods, *SPIRE*, pp. 174–181.

American Heritage Dictionary, www.ahdictionary.com (2011).

Ananthakrishnan, Shankar (2004), Statistical syllabification of English phoneme sequences using supervised and unsupervised algoruthms, *Technical report*, CS562 Term Project Report.

Baayen, R.H., R. Piepenbrock, and L. Gulikers (1996), *CELEX2*, Linguistic Data Consortium, Philadelphia.

Bartlett, Susan, Grzegorz Kondrak, and Colin Cherry (2009), On the syllabification of phonemes, *HLT-NAACL*, The Association for Computational Linguistics, pp. 308–316.

Blainey, Geoffrey (1993), *A Short History of the World*.

Daelemans, Walter and Antal van den Bosch (1992), Generalization performance of backpropagation learning on a syllabification task, *Proceedings of the 3rd Twente Workshop on Language Technology* pp. 27–38, Universiteit Twente, Enschede.

Duanmu, San (2009), *Syllable Structure: The Limits of Variation*, Oxford Linguistics, OUP Oxford.

Goldwater, Sharon and Mark Johnson (2005), Representational bias in unsupervised learning of syllable structure, *Proceedings of the Ninth Conference on Computational Natural Language Learning*, CONLL '05, Association for Computational Linguistics, pp. 112–119.

Goslin, Jeremy (2002), *A Comparison of Theoretical and Human Syllabification*, PhD thesis, University of Sheffield.

Goslin, Jeremy and Ulrich H. Frauenfelder (2001), A comparison of theoretical and human syllabification, *Language and Speech* **44** (4), pp. 409–436.

Hammond, Michael (1997), Parsing syllables: modeling ot computationally, *CoRR*.

Huang, Xuedong, Alex Acero, and Hsiao-Wuen Hon (2001), *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, Prentice Hall PTR, Prentice-Hall, Inc., Upper Saddle River, New Jersey 07458.

Kahn, Daniel (1976), *Syllable-based Generalizations in English Phonology*, PhD thesis, Massachusetts Institute of Technology.

Marchand, Yannick, Connie R. Adsett, and Robert I. Damper (2007), Evaluating automatic syllabification algorithms for English, *6th International Speech Communication Association (ISCA) Workshop on Speech Synthesis*, pp. 316–321.

Marchand, Yannick, Connie R. Adsett, and Robert I. Damper (2009), Automatic syllabification in English: A comparison of different algorithms, *Language and Speech* **52** (1), pp. 1–27.

Mayer, Thomas (2010), Toward a totally unsupervised, language-independent method for the syllabification of written texts, *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, SIGMORPHON '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 63–71.

Muller, Karin (2006), Improving syllabification models with phonotactic knowledge, *Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology at HLT-NAACL*, pp. 11–20.

Ouellet, Pierre and Pierre Dumouchel (2001), Heuristic syllabification and statistical syllable-based modeling for speech input topic identification, *Grammar and NLP*.

Pearson, Steve, Roland Kuhn, Steven Fincke, and Nick Kibre (2000), Automatic methods for lexical stress assignment and syllabification, *INTERSPEECH*, pp. 423–426.

Schmid, Helmut, Bernd Möbius, and Julia Weidenkaff (2007), Tagging syllable boundaries with joint n-gram models, *INTERSPEECH*, pp. 2857–2860.

Selkirk, Elisabeth O. (1984), On the major class features and syllable theory, *in* Aronoff, M. and R. T. Oehrle, editors, *Language Sound Structure*, The MIT Press, pp. 107–136.

Sha, Fei and Fernando C. N. Pereira (2003), Shallow parsing with conditional random fields, *HLT-NAACL*.

Stolcke, Andreas (2002), SRILM - an extensible language modeling toolkit, *INTERSPEECH*.

Wordsmyth Dictionary, www.wordsmyth.net (2011).

Zhang, Jian and Howard J. Hamilton (1997), Learning English syllabification for words, *ISMIS*, pp. 177–186.

Zweig, Geoffrey and Patrick Nguyen (2010), SCARF: a segmental conditional random field toolkit for speech recognition, *INTERSPEECH*, pp. 2858–2861.