Modeling Cue Trading in Human Word Recognition

Louis ten Bosch[1,3] and *Odette Scharenborg*[2,3]

[1] Centre for Language and Speech Technology, Radboud University Nijmegen, The Netherlands
[2] Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands
[3] Donders Institute for Brain, Cognition and Behaviour, Radboud University Nijmegen, The Netherlands

L.tenBosch@let.ru.nl, Odette.Scharenborg@mpi.nl

Abstract

Classical phonetic studies have shown that acoustic-articulatory cues can be interchanged without affecting the resulting phoneme percept ('cue trading'). Cue trading has so far mainly been investigated in the context of phoneme identification. In this study, we investigate cue trading during recognition of *words*, the units of speech through which we communicate. This paper aims to provide a method to quantify cue trading effects by using a computational model of human word recognition. This model takes the acoustic signal as input and represents speech using articulatory feature streams. Importantly, it allows cue trading and underspecification. Its set-up is inspired by the functionality of Fine-Tracker, a recent computational model of human word recognition. This approach makes it possible, for the first time, to quantify cue trading in terms of a trade-off between features and to investigate cue trading in the context of a word recognition task.

Index Terms: cue trading, human word recognition, computational modeling, articulatory features.

1. Introduction

The *cue trading* phenomenon plays a major role in human speech perception. A trading relation between two cues occurs when "... a change in the setting of one cue (which, by itself, would have led to a change in the phonetic percept) can be offset by an opposed change in the setting of another cue so as to maintain the original phonetic percept." ([1], p. 87). Cues that support such a percept can be smeared out in the temporal domain (e.g., nasalization of vowels before nasals), can change in the spectral domain (/p/ before /a/ is spectrally different from /p/ before /i/), or can primarily manifest themselves in the articulatory domain (e.g., articulatory compensation). Moreover, the set of cues that 'make' the percept of a certain phoneme is not unique: i.e., the perception of a phoneme can be based on different combinations of cues; it might be that none of these cues are essential for this phoneme, or that cue combinations can be interchanged with other cue combinations.

Cue trading effects have been extensively shown in classical phonetic studies (e.g., [2], [3]), which mostly focused on phoneme identification in stimuli ranging in duration from short phoneme sequences up to sentences. But these effects also play a major role in recent technological applications. For example, cue trading effects are of immediate practical importance for optimization schemes of hearing devices (e.g., see [4]). The novelty of our approach is that the cue trading phenomenon is studied in the context of a *word* recognition framework rather than pure phoneme identification. This approach addresses one of the open problems in human speech perception: exactly how cues are traded and integrated to support the perception of a certain speech unit in the context of the recognition of words. The aim of this study is to shed light onto this issue by investigating cue trading in a computational model of human spoken-word recognition. To that end, we address three research questions: 1) how can cue trading be quantitatively dealt with within a model; 2) how can trading relations be found in an automatic way within the paradigm of word recognition, and 3) how to describe the relationship of cue trading with the temporal dynamics of features?

Since the speech signal is not a sequence of discrete invariable units but is characterized by articulatory anticipation, coarticulation, and assimilation, cue trading goes hand-in-hand with asynchronous transitions of features varying over time (see also [5],[6]). For example, nasalization of vowels preceding /n/ may be described by an early rise (already during the vowel) of the feature *nasality* (i.e., an effect in the temporal domain); at the same time this leads to spectral changes in the vowel and so to a trade-off between the features present at a specific moment in the nasalized vowel allophone, compared to its non-nasalized variant.

Albeit not under the same term, cue trading also plays a role in automatic speech recognition (ASR). There, cue trading is actually closely related to cue *weighting*. During training of the ASR model, the training algorithm adjusts the ASR model parameters in order to optimize the likelihood of the speech training data given the model. A number of these model parameters are specifically used to weigh features, since they are applied in a weighted sum of feature values that is used in the "goodness of fit" (in terms of the likelihood) between signal and model. In this paper, we will make use of the fact that ASR systems are able to automatically adapt such weightings via the model training (see following sections).

In our model, we will use articulatory features (AFs) as cues to represent the speech signal. AFs describe the speech signal in terms of estimated values of speech production parameters, e.g., *manner* and *place of articulation*, *tongue height*,and *lip rounding* – often inspired by [7] (see e.g., [8]). Our motivation to use AFs is two-fold. Firstly, AFs have already proven useful in the computational modeling of the prelexical level in human spoken-word recognition [9]. Moreover, AFs are not constrained to change synchronously, and therefore allow addressing possible asynchronicity between features. Table 1 lists the articulatory features used in this study.

2. The computational model

In line with [9], our computational model assumes that the speech recognition process consists of a prelexical level and a lexical level. First, listeners map the incoming acoustic signal onto so-called prelexical representations (i.e., AFs, see Section 2.1). At the lexical level, all lexical representations are stored in the form of sequences of AFs, and lexical representations that (partly) match the prelexical representations are activated in parallel (see Section 2.2).

Table 1. Specification of the AFs; nil is a code for non-applicable.

Articulatory Feature	AF values
manner	plosive, fricative, nasal, glide, liquid, vowel, retroflex, silence
place	bilabial, labiodental, alveolar, palatal, velar, glottal, nil, silence
voice	+voice, -voice
front-back	front, central, back, nil
round	+round, -round, nil
height	high, mid, low, nil
duration-diphthong	long, short, diphthong, silence

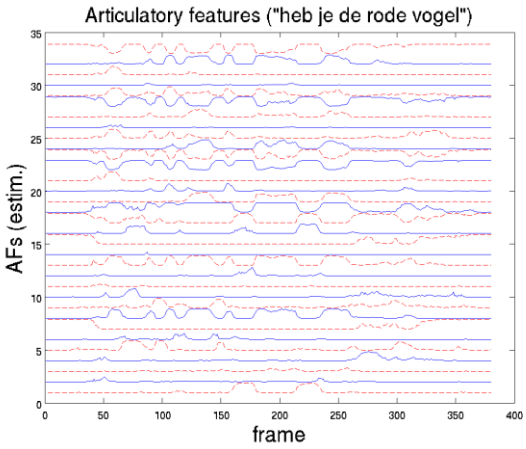

Figure 1. The 33 AF estimates for a Dutch sentence ('Do you have the red bird?'). The horizontal axis represents time (in terms of frames). For the sake of clarity, the AFs have not been labeled separately.

2.1. The prelexical level

The prelexical level consists of seven artificial neural networks (ANNs) used in parallel. Each ANN was trained for one of the seven AFs groups using the NICO Toolkit [10], an artificial neural network toolkit designed for speech applications. For training the ANNs, 3410 randomly selected utterances from the manually transcribed read speech part of the Spoken Dutch Corpus [11] were used. These data were labeled at the phoneme level, and prior to training of the ANNs converted to their canonical AF value representation using a fixed phoneme-AF value translation table. Each ANN consists of an input, a hidden, and an output layer. The output layer presents an estimate for each of the AF values for that particular AF (see Table 1), with estimates between 0 (property absent) and 1 (present). An example of the estimates for all 33 AF values in the prelexical level over time for the utterance "Heb je de rode vogel?" (*Do you have the red bird?*) is displayed in Figure 1. The horizontal axis represents time (frame index), while the 33 AF value estimates are presented along the vertical axis. The leading and trailing silences are clearly visible in the figure. For the sake of clarity, the 33 AF values have not been labeled separately.

2.2. The lexical level

The implementation of the lexical level in our computational model consists of a hidden Markov Model (HMM) recognition system (based on HTK [12]) – a conventional technique in ASR. The set-up of the model, however, is largely inspired by the architecture in Fine-Tracker, a recently developed computational model of human word processing [9]. Because of our model's embedding in an ASR framework, it has the additional option of optimizing parameters based on real speech.

Following Fine-Tracker, in our model, AFs can change asynchronously in time at the prelexical level and at the lexical level are mapped in a left-to-right fashion onto sequences of 'lexical' AF vectors. Each lexical vector consists of 33 AF values. Each value is either an 'ideal' canonical target value or is left unspecified ('underspecification'). For example, for /a/ only those AFs that differentiate the /a/ are specified, while other AFs are left unspecified: e.g., *manner*:vowel = 1, *manner*:silence = unspecified, while all other values of *manner* are 0; for the feature *voice*:–voice = 0, *voice*:+voice = 1; etc.

In Fine-Tracker, the lexical level is represented by a word search module. This word search module uses a probabilistic word search to match the prelexical feature vectors with the candidate words in the lexicon in order to find the most likely sequence of words. For each of the prelexical vectors the "goodness of fit" (GOF) with the lexical vector is calculated, a worse fit results in a lower 'activation' of that word and vice versa. The 'unspecified' values in the lexical vectors are *ignored* in the calculation of the GOF between the input AF vectors and the lexical AF vectors (note that there are no 'unspecified' components in the vectors created by the prelexical level as these vectors are created by the ANNs). The GOF in Fine-Tracker weights the features *equally* within the set of specified features.

Inspired by Fine-Tracker's implementation, our HMM-based model also deals with underspecification. This is done by representing each lexical vector by a single-state HMM with a single Gaussian distribution (with diagonal covariance matrix) as follows. Because each lexical vector has 33 components, its Gaussian is characterized by 33 means and 33 variances. First, each *specified* value in the Fine-Tracker lexical vector is directly used as the corresponding mean in the Gaussian; the remaining mean values are set to 0.5. Secondly, for each specified AF value in the lexical vector, the *variance* of the Gaussian is set to a small value (0.05), while for the unspecified AF values the variance is set to 0.4. In this study, the variances are the important parameters, since the inverse of the variance determines the AF weight used in the GOF between the prelexical AF value and the HMM. Therefore variances fully determine the trade-off between the features. The 8 times larger variance for *un*specified components imply that these components hardly matter in the calculation of the GOF compared to the specified components.

This architecture addresses our first research question by showing how cue trading can be dealt with in a computational model. The mean and variance settings described here will be referred to as the *baseline* settings.

3. Methodology

3.1. Experimental set-up

Our second research question concerns how cue trading relations can be found in an automatic way within the paradigm of word recognition. This is addressed by the ASR training step. In this research, this training step is specifically constrained to *only* update the variances; the means remain fixed. Starting from the baseline settings, each training iteration will lead to updated variances, which in general will be different for each AF and each Gaussian (i.e., each phoneme). The ASR model training is done via Expectation Maximization (EM), a conventional way to optimize model parameters in ASR [12].

The model will be evaluated on the basis of its recognition performance measured by means of Word Error Rates (WER) (Section 4). If the word recognition performance can be improved through the adaptation of the lexical vector HMMs, this signals that cue trading takes place (because the variances are the only parameters that can be adjusted).

Our third research question will be addressed in section 5, where we analyze the resulting cue trading by studying the feature weights per AF and the relation with feature asynchrony.

3.2. Material

3.2.1. Test set

The test set used in our study is taken from a speech database that was developed as part of the ACORNS project [13], which aimed at investigating and modeling language acquisition by young infants. Therefore, all utterances use a small lexicon and have a simple syntax, similar to child-directed speech. There are 83 different words in the lexicon. The total number of speakers is 10 (4 females, 6 males). The average duration of the speech files in the test set is 5.4 s.

The data chosen for the experiments consisted of a set of 5986 utterances from the Dutch part of the ACORNS database. 5386 of these utterances were used for adjusting the Gaussian variances, while 600 utterances were used for testing the model and investigating cue trading.

3.2.2. Language model

It is well known that word frequency and the context of a word play a role in human word processing. We therefore applied a straightforward bigram (containing 83 unigrams and 688 bigrams) that was built on a set of 4490 utterances (disjoint from the training and test sets) with the same syntactical structure as the test utterances used in the experiments. The test set perplexity of this bigram was 20.1. There are no out-of-grammar words in the test set.

4. Word recognition results

The results obtained by our model are presented (in percentages) in Table 2. The final column presents Word Error Rates (WER). For the sake of completeness, the table also shows the word accuracy, substitutions, deletions and insertions (denoted Acc, Subs, Del, Ins, respectively). The row 'Baseline' presents results obtained with the baseline settings of the lexical vector HMMs while the rows indicated with 'After N iteration(s)' show the word recognition results of our model after training the model for N iterations. After 5 iterations, the performance has stabilized.

The improvement of the model (compared to the baseline model) shows that cue trading indeed takes place, and vice versa: this trading indeed helps to improve the recognition performance.

5. Feature weighting and asynchrony

5.1. Feature weighting per AF

While in the baseline lexical model all specified AF components have equal weight, this is no longer the case after training: each lexical AF vector HMM model drifts away from its initial setting and is thereby updated in its own (phoneme-specific) way.

Figure 2 shows the effect of the automatic adaptation of the cue weights, split out per AF. It displays the weight of each of the 33 individual AF values (along the horizontal axis), for the baseline model and the models after 1 to 5 training iterations, averaged across all lexical vectors (i.e., phonemes). In this averaging, the silence and short pause (sp) model were excluded in order to focus on the 'real' speech models.

For clarity, all weights are normalized such that the baseline model corresponds to the constant value 1 (represented by the horizontal dashed curve). The other curves show that each subsequent iteration makes the weights more and more pronounced. Comparing Figure 2 with Table 1, Figure 2 clearly shows the emerging relevance of the *manner* and *place* features in comparison to the other features, evidenced by the higher weights for the *manner* and *place* AF values. This means that, among all AFs used, the *manner* and *place* features help most to distinguish phonemes from each other within this word recognition task. Moreover, Figure 2 reveals differences between the weights *within* the group of each AF. The 'nil' and 'silence' components correspond to the valleys in the plots - they hardly gain weight during further training. This reflects that these AFs never become relevant in the decision which word has been produced.

Table 2. Performance of the computational model on a 3450-word recognition task.

	Acc	**Subs**	**Del**	**Ins**	**WER**
Baseline LM	80.95	16.79	2.26	2.28	21.33
After 1 iteration	83.17	14.82	2.01	2.12	18.95
After 3 iterations	89.28	8.76	1.96	1.84	12.56
After 5 iterations	89.58	8.44	1.98	1.83	12.25

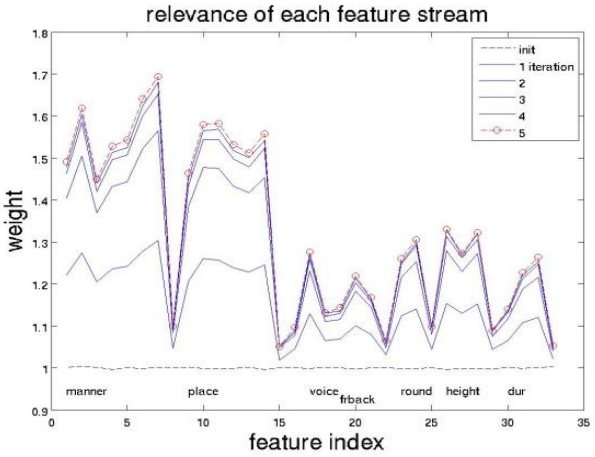

Figure 2. Weightings of the AF values, shown for the baseline model (dashed curve) and after N training iterations. The ordering of the AFs is the same as in Table 1.

5.2. Relation with feature asynchrony

To address our third research question, i.e., how cue trading relates to the temporal dynamics of features, we analyzed AFs in terms of their temporal organization using a method motivated by findings about feature asynchrony [14]. To that end, the training corpus was first automatically aligned (via forced alignment) by using the baseline models with the 'canonical' phoneme representations. Next, we select an AF value (e.g., *plosive*) and we compare the time course of the *plosive* AF value (as measured in the speech signal by the ANNs) with the position of the frames of a *plosive* (e.g., for /p/) as specified by the forced alignment.

Figure 3 provides an example of this analysis. For the sake of transparency, we focus on three *manner* features (since *manner* AF values undergo prominent weight updates, as demonstrated in the previous section), namely *plos(ive)*,

nas(ality), and *fric(ative)*. The fourth curve (*ave*(*rage*)) shows the average time course of all AF values. Each curve is calibrated in the x-direction such that the center frame in the curve coincides with the first frame in the speech signal that was assigned to the canonical lexical vector.

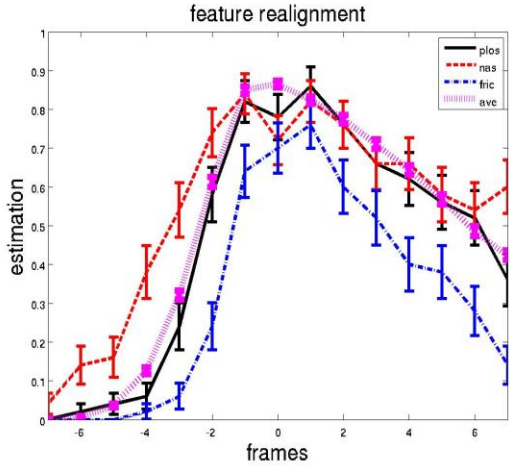

Figure 3. The time course of three manner AF values (plos(ive), nas(al), fric(ative)) after synchronization via forced alignment with the canonical lexical representation. The horizontal axis displays time (in frames), while the vertical axis shows the average estimation of the AF value. The thicker curve shows the average (ave) curve across all AFs. (Best seen in color.)

Figure 3 clearly shows differences in the dynamic behavior of these AFs. For example, the (red) dashed curve, which represents *nasality*, shows, compared to the three other curves, an early rise, while *fricative* (blue dashed) shows a late rise and a lower peak (around 0.73, compared to 0.85 for the average curve). As can be seen in Figure 3, none of the AFs reaches value 1 at the center frame (at x=0), showing that the actual AF value at the prelexical stage often 'undershoots' the target canonical lexical AF values in the lexicon.

The non-parametric Kolmogorov-Smirnov test shows that the three displayed manner features (i.e., *plosive*, *nasality*, and *fricative*) differ significantly from the average contour (all *p*s < 0.01). Figure 3 shows that, among the *manner* features, the *nasality* feature undergoes the most prominent effect of articulatory anticipation.

6. Discussion and conclusion

In this work, we addressed three research questions: 1) how can cue trading be quantitatively dealt with within a computational model; 2) how can cue trading relations be found in an automatic way within the paradigm of word recognition, and 3) how to describe the relation between cue trading and the temporal dynamics of features. These questions were addressed by investigating cue trading in a computational model of human word recognition. The main finding is that the word recognition performance of the model using the baseline settings for the AF weights can be improved by automatic adjustment of these weights on the basis of real speech data, showing that cue weighting takes place and that cue trading relations can be found automatically.

Figure 2 shows that *manner* and *place* are the most relevant AFs for word recognition; it further shows different weights *within* each group of AFs (especially *manner*). These differences across AFs reflect the different distinctive 'power' of AF values to distinguish words in a recognition task. The 'nil' and 'silence' components in the AF specifications do not gain much weight by further training: These features do not carry *distinctive* information for word recognition.

Because each HMM represents a different phoneme, our computational model not only generates feature-dependent weightings but is also able to produce *phoneme*-dependent cue weightings. In the near future, we will investigate how cue trading differs across phonemes. We will further deepen the relationship (pointed out in Section 5) between the feature weighting on the one hand and the feature asynchrony on the other. This refined analysis will require more speech data than used in our experiments here to avoid data sparseness.

For HSR research, these results show that ASR-based speech analysis methods inspired by knowledge about human speech processing can be of great value to investigate properties of speech on a large speech corpus. The availability of models like the one proposed here is an indispensible asset to narrow the gap between computational models of human speech processing on the one hand, and ASR-based speech analysis methods on the other. As a proof of this, we plan to apply the cue weightings found in section 5 to refine the goodness of fit mechanism in Fine-Tracker [9].

7. Acknowledgements

The research by Louis ten Bosch is partly sponsored by the European FP7 OPTIFOX project (nr 262266). The research by Odette Scharenborg is sponsored by the Max Planck International Research Network on Aging.

8. References

[1] Repp, B.H., "Phonetic trading relations and context effects: New experimental evidence for a speech mode of perception", Psychological Bulletin 92, 81-110, 1982.

[2] Howell, P., "Cue trading in the production and perception of vowel stress", J. Acoust. Soc. Am. 94(4), 2063-2073, 1993.

[3] Kewley-Port, D., Zheng, Y., "Vowel formant discrimination: Towards more ordinary listening conditions", J. Acoust. Soc. Am. 106, 2945-2958, 1999.

[4] Winn, M.B., Chatterjee, M., Idsardi W.J., "The use of acoustic cues for phonetic identification: Effects on spectral degradation and electric hearing", J. Acoust. Soc. Am. 131(2), 1465-1479, 2011.

[5] Cohen, J.R., "Segmenting speech using dynamic programming", J. Acoust. Soc. Am. 69(5), 1430-1438, 1981.

[6] Ostendorf, M., "Moving beyond the 'beads-on-a-string' model of speech," Proc. IEEE ASRU Workshop, 1999.

[7] Chomsky, N., Halle, N., "The Sound Pattern of English", Harper and Row, 1968.

[8] King S., Taylor, P., "Detection of phonological features in continuous speech using neural networks", Computer Speech and Language 14(4), 333-353, 2000.

[9] Scharenborg, O., "Modeling the use of durational information in human spoken-word recognition", J. Acoust. Soc. Am. 127 (6), 3758-3770, 2010.

[10] Ström, N., "Phoneme probability estimation with dynamic sparsely connected artificial neural networks", The Free Speech Journal 5, 1-41, 1997.

[11] Oostdijk, N.H.J., Goedertier, W., Van Eynde, F., Boves, L., Martens, J.-P., Moortgat, M., Baayen, H., "Experiences from the Spoken Dutch Corpus project", Proc. LREC - Third International Conference on Language Resources and Evaluation, Las Palmas de Gran Canaria, 340-347, 2002.

[12] Young, S., et al., *The HTK Book*. Version 3.4, March 2006.

[13] ten Bosch, L., Van hamme, H., Boves, L., Moore, R.K., "A computational model of language acquisition: the emergence of words", Fundamenta Informaticae 90, 229-249, 2009.

[14] Frankel, J., Wester, M., King, S., "Articulatory feature recognition using Dynamic Bayesian Networks", Computer Speech & Language 21(4), 620-640, 2007.