25. Новые слова и значения: по материалам прессы и литературы 80-х гг. – М., 1997.

26. Толковый словарь русского языка конца XX столетия: Языковые изменения / под ред. Г.Н. Скляревской. – СПб, 1998.

27. Толковый словарь русского языка: в 4 т. / под ред. Д.Н. Ушакова. – М., 1935-1940.

Hagen Peukert,
Prof. Dr. Josef Wallmannsberger

FREQUENCY AND SIMILARITY: A STATISTICAL APPROACH TO THE DEVELOPMENT OF THE LEXICON

Department of General and Computational Linguistics,
Kassel University

As we have shown elsewhere [10], detecting word boundaries using transitional probabilities between speech units can be substantially improved by deploying additional statistical information in the text, such as the most frequent phoneme chains. This idea also simulates a possible interplay between bottom-up and top-down processes whereas the top-down information was extracted in the previous bottom-up calculations. We have further improved the algorithm and now like to present the results of two issues. First, how do recursive structures influence the outcome of the segmentation process depended on the size of the sample and, second, how can we use this information in building a lexicon. In a wider sense, both questions are related to language acquisition, which is seen as a meta-model for copying efficient processes of automatic language learning.

The algorithm used in [10] calculated all transitional probabilities occurring between one to five phonemes of a corpus sample and inserted whitespaces at a predefined limit running from 0 to 1. This value was defined as a frequency ratio of the total occurrences of the entire phoneme chain and the occurrences of its subparts. Moreover, the five most frequent n-grams were taken from the text and we gave evidence that these n-grams carry additional cues for word segmentation since most of the time they happen to be words if merged into larger sets, thus, each defining two more word boundaries if thrown back into the corpus. The most recent version of the algorithm has three alterations. First, it does not take the most frequent n-grams of the corpus sample, but calculates a predefined number of the words that are separated from the transitional probabilities calculations. These words are saved in a list, which represents some preliminary lexicon. Second, the recursive structure can be defined by the

number of repetitions. This allows, in a figurative sense, to simulate stages in the development of the lexicon acquisition and therefore also for the sequence of events throughout time. Third, for each cycle one may chose to input different corpora. From the perspective of language acquisition research, this architecture is somewhat closer to reality because one could set fourth that the young language learner will have different speech input throughout the first months of life as well.

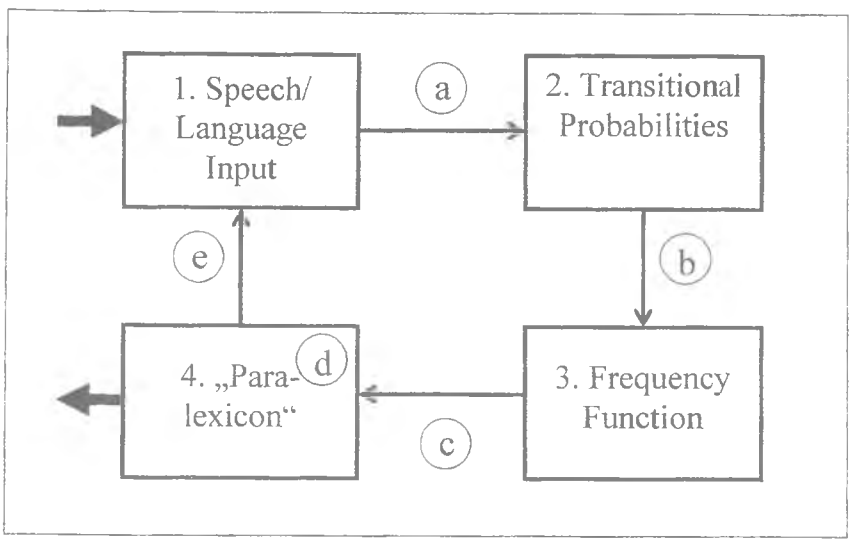

Figure 1. General Design of updated Algorithm

The letters a through e indicate the psychological processes and experimental findings as follows: a – Phonemes are perceived categorically and are the perceptual units in English (e.g.: [9], [2], [3], [12]), b – 8 month olds can use transitional probabilities to segment an unknown speech stream [11], c – infants recognize most frequent sound chains [4], d – sound sequences can be memorized for some time (e.g. [5], [6]), e – phonemic representations can be aligned in a top-down mechanism [1].

Having input child directed speech from the CHILDES database [7], converted it to a machine readable phonetic transcription and removed whitespaces, the algorithm starts to compute all transitional probabilities between the given segments varying in length from one to five. An outer loop running from zero to one inserts whitespaces whenever the transitional probability amounts to less than the respective value in the outer loop.

Depending on a value ζ defining the number of entries in the paralexicon[1], the ζ most frequent chains segmented by the procedure just outlined are stored in a list. A last loop that is put over on the very top of the algorithm so far specifies the number of recursions. At the beginning of each cycle, the stored lexicon entries are matched with its counterparts in the same or a different corpus adding for each entry two more whitespaces to the input text. The matching process follows the same assumptions made in the cohort model [8]. Then, the first and second loop starts to work enriching the lexicon list with new entries for every run through the loop.

The performance of the segmentation is output as F1-measure.[2] The F1-maxima at constant phoneme length, frequency ratio and corpus size can be described in dependence of the recursive cycles and the size of the lexicon (figure 2). As we have expected, the size of the lexicon if greater then ten entries does not affect the performance of the segmentation substantially for corpora smaller than 10000 phonemes (about 2500 words). It will, however, have a tremendous effect for larger corpora and long phoneme chains. Here performance may double as the example in our largest corpus shows. Thus, the larger the corpus and the phoneme chain becomes, the more important is the size of the lexicon for a successful segmentation performance. The number of iterations will also bear positive effects on segmentation results. Independent of the size of the input, every corpus will increase its F1-measure by a considerable amount for the first cycle. For smaller lexica it will then continue to rise slightly or stay constant. Longer phoneme chains will profit more with each additional run. For large corpora, smaller phoneme chains may even decrease if ζ becomes bigger.

---

[1] We chose ζ to be 10, 20 or 30 entries respectively because preliminary tests had shown them to be critical lexicon sizes.

[2] For reasons of comparability with all other major contributions in that area, we decided to use the F1-measure used in the field of Information Retrieval. It is a special form of the harmonic mean and as such defined as

$$F_\beta = \frac{1+\beta^2}{\frac{\beta^2}{r}+\frac{1}{p}} \text{ and } \beta \geq 0.$$

whereas r stands for recall and p for precision, which are defined as

$$precision = \frac{number \cdot of \cdot correctly \cdot calculated \cdot whitespaces}{number \cdot of \cdot inserted \cdot whitespaces}$$

$$recall = \frac{number \cdot of \cdot correctly \cdot calculated \cdot whitespaces}{number \cdot of \cdot whitespaces \cdot in \cdot the \cdot corpus}$$

We set β = 1 and hence weighted r and p equally.

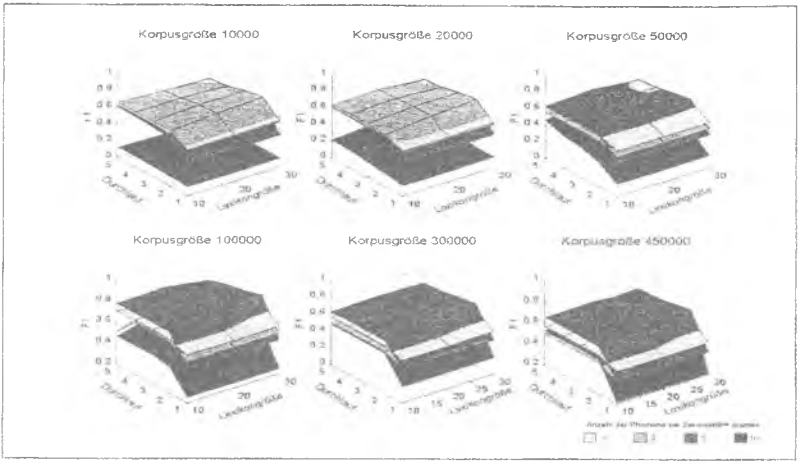

Figure 2. Segmentation performance (F1) dependent on the number of cycles, ζ and length of the phoneme chain for six different corpus sizes (the number next to "corpus size" indicates the number of signs in the corpus)

Our simulations give evidence for noticeable improvements in the segmentation outcome if using simple procedures recursively including additional statistically encoded information in the text. We run our simulations in two modes. First, we input only one corpus for all cycles and, second, we used different corpora for each cycle. The results in both modes were very similar. Only minor improvements (an average of 3%) could be observed. So, at first glance, it seems that different corpora do not impact on correct segmentations. This is somewhat counterintuitive. One would assume that new corpora would at least generate some novel words to the lexicon and so add new information for the next iteration. Looking at figure 2, it is apparent that after the first iteration, F1 doubles and thereafter increases, if at all, only little. Further investigation of the lexicon entries revealed the main source of the problem. It turned out to be, of course, the lexical embeddedness of bound morphemes and articles (some entries are subparts of other entries). This problem will be particularly interesting after the second iteration since entries that do not fit the pattern of transitional probabilities are also present in the corpus by then.[1]

[1] The rationale behind it is clear cut and also stochastically plausible. After the first iteration, the probability of encountering an embedded word that is followed by a sequence of signs that is also enlisted in the lexicon is low. This is compelling because these very entries, by definition of transitional probabilities, are enlisted for their quality of exactly this sequence of signs. After the second iteration this is not necessarily true because now there are entries in the lexicon that were not segmented by transitional probabilities.

To minimize this shortcoming, we changed the algorithm but at one point. We would not allow the lexicon list to be matched with the input stream for a certain number of cycles. As a consequence, the lexicon list would now grow almost linearly by a certain number of words (ζ) until the specified number of cycles is reached. Then it will input them at once and from this moment on will continue the algorithm as usual, that is, matching the lexicon with each cycle to come. In our simulation we used 30 corpora containing 10000 signs and set ζ equal to ten. As we have hypothesized, independent of the number of corpora or cycles respectively, the improvement of F1 (Δ F1) will decrease substantially after the second iteration (figure 3). However, for each additional iteration, in which matching was delayed, the performance rose by some percentage. This showed that, indeed, every new corpus also adds some more information that is useful for segmentation. Furthermore, the higher the delay of the lexical matching process (the more cycles run through without giving the information of the lexicon list), the more the slope of the lexical function (LexFunc) will smooth out. The maximum values of this family of functions define another function (argmax(LexFunc)) whose first derivative runs against zero. From this we may conclude that for an infinite number of cycles (i), the lexical function will converge to argmax(LexFunc) and at some point be equal to it ($i \to \infty$, LexFunc = argmax(LexFunc)). In what follows, we will explain these interesting findings.

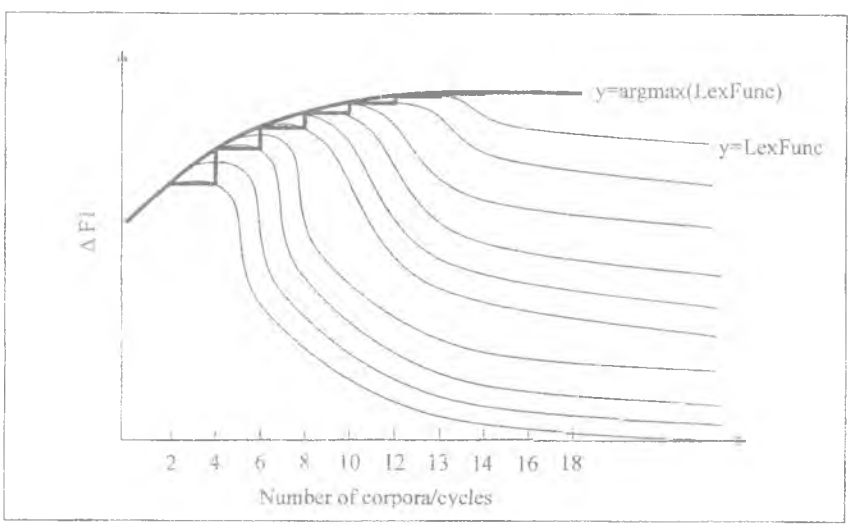

Figure 3. Schematic depiction of family of parametric lexicon functions

We will now take a closer look at the sudden but regular decrease after the second iteration that arise from the moment the lexicon matches are switched on. The weakness of the transitional-probability-approach is that, depending on the specific sound environment, the same word is sometimes correctly separated and at other times not. Then it is part of a larger phrase or even torn apart. Still, the correct segmentations of that word will be more frequent. It is this observation that led us to design our algorithm to compensate the deficiency of transitional probabilities. As an extreme case for example, the word /mʊmi/ (mommy) may appear ten times in the corpus in different sound environments. Even if it is only separated twice as /mʊmi/, it will still be written to the lexicon as long as the ζth entry is not greater than two. One of the advantages of using transitional probabilities, on the other hand, is that the algorithm also recognizes co-occurrences and separates them into its distinct parts, of course only if the sample is large enough to contain different sound environments, in which the subparts of the co-occurrences also show up. Typically this is the case for article – noun sequences. Since the article does not only occur before the same noun but different nouns and as well in other phrases (e.g. of the best; the beginning, etc.), the definite article will be correctly separated by transitional probabilities and because of its high frequency also be present in the lexicon list. With the exception of the Progressive /ɪŋ/ (ing), grammatical morphemes (Puralallomorphs [s], [z] and Pastallomorphs [t], [d]) and other bound morphemes do not occur in the lexicon list after the first iteration. They appear in such sound environments which add them to the next word or they are not separated from the word they belong to. Now the second iteration starts with the matching process and the lexicon entries are aligned with the new corpus. The effect is positive because all frequent chains that were segmented at one time but not at another, will now be consistently filled in. In the above example, all /mʊmi/-words would then be correctly separated or strung together respectively, which betters the result by a factor of at least five in this case.[1] In addition to that, some more information is added because predecessor and successor words also receive at least one correct word boundary. Applied to the /mʊmi/ - example, we have a surrounding context that could look like /ðɪs ɪz mʊmi switi/ (this is mami, sweetie) and was segmented into /ðɪs ɪzmʊm is witi/. Filling in /mʊmɪ/ not only results in two more correct counts of whitespaces, but also for deleting a wrong whitespace within the corresponding word. In fact, it also leads to a change of the probability distribution within the successor word, which might now possibly be separated correctly depending on the other environments

[1] Taking the wrong segmentation into account, the factor would increase further because in the worst case each word might contain three more whitespaces, which totals up to 24 wrong segmentations.

of the sound chain /switi/ in the given material. This is so because the /i/-/s/ and /s/-/w/ transitional probabilities will be set to the corresponding value of the second outer loop. Thus, for the specific environment /mɒmi switi/ the algorithm has "learned" that a whitespace between /mɒmi/ and /switi/ is more likely than stringing together the /i/ and /s/, which is still remembered as "valid" for most other environment since /ɪs/ is a very frequent phoneme chain. These processes even lead to the positive result of segmenting the indefinite article /ʌ/ (a) correctly. As opposed to the definite article /ðʌ/, the indefinite article only consists of one phoneme and occurs in such a large variety of different sounds that it almost never stands by itself.

However, the positive effect turns negative soon after the second iteration, especially for cycles comprising a short lexicon list. The problem is the bound morphemes; grammatical morphemes in particular. A lexicon list will contain the most frequent nouns and verbs of the corpus (e.g. ball, mommy, daddy, spoon, let, sit, get, do, are, go). They occurred in different sound environments and are therefore seen as units. Indeed, in some of the environments, they are used in the Plural, Progressive, 3rd Person or Past. While it does not state a problem for irregular forms, plurals as well as past,[1] it is a challenge for regular forms. Only a few Plurals and 3rd Person forms[2] suffice to separate the grammatical morphemes from its root. To be more precise, the entry /bɔl/ will cut apart the /z/ from its plural form. By the same process, /sɪts/ and /gets/ lose their /s/ as well if only /sɪt/ and /get/ are enlisted in the lexicon. As a result, the /s/ and /z/ will stand alone as atomic and independent items. As such, they will be counted as all other potential candidates for the next lexicon list and they have very good chances to be in there. The same happens with the progressive and some other bound morphemes. Finally, the third iteration starts and the new list is matched. In line with the cohort model, each item from the lexicon list will now be aligned with the possible candidates in the new corpus activating the longest phoneme chains that fit the target chain. Whenever, some longer phoneme chain has no correspondent in the lexicon, smaller units such as plural and past indicators will be inserted giving two wrong whitespaces. For example, /sʌspekt/ (suspect) and /sɪzɜz/ (scissors) are very unlikely to be in the lexicon. In the first case, the entry starts with the /s/ and having no other entry /sʌ/ in the list, the candidate turns to /s ʌ s pekt/. Assuming that neither /pek/ nor its subparts are enlisted, the next entry is the grammatical morpheme /t/ that will be merged. At the end, the word is segmented into five potential entries for the

[1] Most of the time they are segmented as entire units.

[2] Naturally, if they appear together more often, they are attached to the lexicon entry and the problem does not come up.

lexicon. "Scissors" would even be divided into /s ɪ z ɜ z/ provided that there are no entries for /sɪ/, /zɜ/ and /ɜz/ in the list. The latter is especially tricky because now there are a large number of single phonemes in the text that have easy access to the lexicon for their high frequency occurrences. (here /ɜ/ and /ɪ/).

While the same processes have already taken place with the /ɪŋ/ forms in the first and second iteration, from the third iteration on, the effect of additional single phonemes in the lexicon will outweigh the positive effects of the recursive structures if the lexicon list is too small. The negative effect is much less intense if the lexicon reached a certain size because then there will be fewer phoneme chains that cannot be matched with the lexicon. This explains the smoothing out of the slope in figure 3 as time (cycles) progresses.

So, could one question the recursive structure at all? The argument goes that the lexicon list will build up infinitively as long as new input prevails until enough words are available that the text can be entirely segmented. This is, however, not the case as corpus statistics evidently suggest. With every cycle the number of entries will concavely decrease (figure 4). Evidently, the most frequent phoneme chains will appear in every corpus. As a consequence, with each new corpus fewer new chains will be added, so that the total number in the lexicon will be less than 100 entries independent of ζ even after 30 cycles.

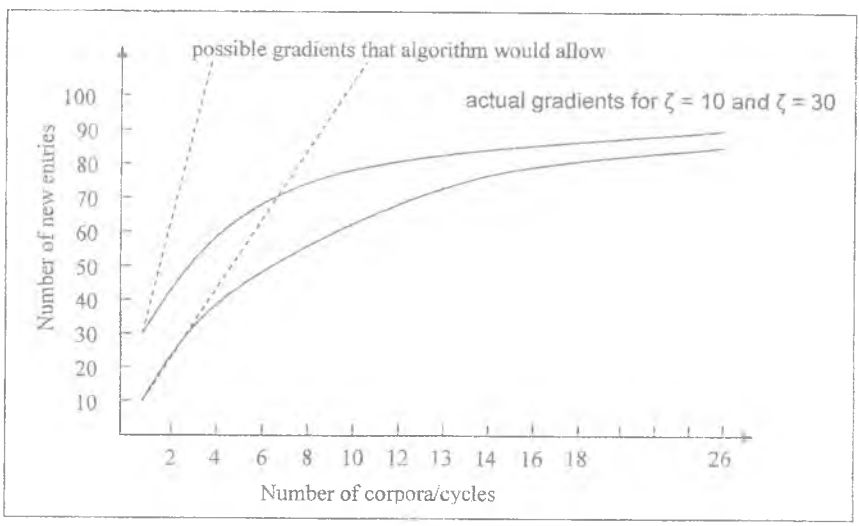

Figure 4. Schematic depiction of lexicon growth for a 10000 sign corpus

Our research on statistical word segmentation contributes to two burning issues in the linguistic community. First, the bootstrapping problem has an

alternative and plausible solution that does not take the loss of contradictory theoretical, experimental and logical findings. Neither semantics assuming syntactic structures that can only be build up knowing some semantics nor vice versa is necessary to construct an initial list of words as a pattern from which all other principles and rules can be derived. All that is needed is the ability to use transitional probabilities to segment a speech stream and recognize the most frequent candidates from there. Both abilities are experimentally proven. Second, our simulations suggest that the lexicon should be structured morphemically. Once we admit recursive structures, which compensates the weakness of transitional probabilities, a morphemic segmentation is inevitable. The simulation discloses morphemes as robust building blocks of language within a statistical framework of language learning.

Bibliography

1. Bortfeld, Heather; James L. Morgan; Roberta Micknick Golinkoff; Karen Rathburn (2005): Mommy and Me: Familiar Names Help Launch Babies Into Speech-Stream Segmentation, American Psychological Society, Vol. 16 (4), 298-304.

2. Brent, Michael R. (1999): An efficient, probabilistically sound algorithm for segmentation and word discovery, in: Machine Learning, 34, 71-105.

3. Jacobs, Arthur M. (2003): Simulative Methoden, in: Gert Rickheit; Theo Herrmann; Werner Deutsch (Ed. Issue 24). Psycholinguistics: An international Handbook, Ernst Wiegand (Ed.) Handbooks of Linguistics and Communication Science, Berlin, de Gruyter, 125-142.

4. Jusczyk, Peter W.; Paul A. Luce; Jan Charles-Luce (1994): Infants' Sensitivity to Phonotactic Patterns in the Native Language, in: Journal of Memory and Language 33, 630-645.

5. Jusczyk, Peter W.; Richard N. Aslin (1995): Infants' detection of the sound patterns of words in fluent speech, in: Cognitive Psychology, 29, 1-23.

6. Jusczyk, Peter W.; Elizabeth A. Hohne (1997): Infants' Memory for Spoken Words, in: Science (277), 1984-1986.

7. MacWhinney, Brian (1995): The CHILDES Project: Tools for Analyzing Talk, Erlbaum, Hillsdale.

8. Marslen-Wilson, William D.; Alan Welsh (1978): Processing Interaction and Lexical Access during Recognition of Continuous Speech, in: Cognitive Psychology, 10, 29-63.

9. Pisoni, David B.; Paul A. Luce (1987): Acoustic-phonetic representations in word recognition, in: Cognition, 25, 21-52.

10. Wallmannsberger, Josef; Hagen Peukert (2006): Word Boundary Detection Based on Distributional cues, in: Discourse Space of Communication in Modern Germanic Languages, International Collection of Linguistic Papers, Samara University Publishing House. Samara, 115-121.

11. Saffran, Jenny R.; Richard N. Aslin; Elissa L. Newport (1996a): Statistical learning by 8-month-old infants, in: Science (274), 1926-1928.

12. Swingley, Daniel (2005): Statistical Learning and the contents of the infant vocabulary, in: Cognitive Psychology, 50, 86-132.