

Lexicon Optimization: Maximizing Lexical Coverage in Speech Recognition through Automated Compounding

Vincent Vandeghinste

Centre for Computational Linguistics
Maria Theresiastraat 21, 3000 Leuven, Belgium
vincent.vandeghinste@ccl.kuleuven.ac.be

Abstract

In this report we show that a lexicon can be designed in such a way that lexical coverage can be maximized by real-time lexicon expansion and a limited word part lexicon for Dutch speech recognition. More specifically, we describe how the lexicon is designed and how the real-time expansion module was built and tested. Tests were performed using a 36.000 entries lexicon. The test results show that out-of-vocabulary rates are rather small, due to automated rule-based compounding of the lexical building blocks. Statistical information was included to improve the accuracy of the rule-based compounding system. This approach proved to be successful.

1. Introduction

When designing a speech recognition system for continuous spontaneous speech, it should be taken into account that, even with a lexicon size of approximately 40.000 entries, a number of words uttered by the speaker will not be amongst those entries.

It should also be taken into account that Dutch is a productive language: Content words form an open class. New words can be formed on the basis of already existing words, they can be taken from another language, or they can result from a combination of a foreign word with Dutch affixation. Language users are capable of forming new words by using the relations between the newly formed words and their base forms (Booij & van Santen, 1995). A number of these relations are systematic, and can hence be captured by an automated system.

Designing a lexicon where the lexical coverage is optimized for a fixed vocabulary size is an important aspect in lexicon development for large vocabulary speech recognition systems. Minimizing the out-of-vocabulary (OOV) rate contributes to minimizing the global error rate of a recognition system. Different estimates of the number of recognition errors due to each OOV-word range from 1.2 to 2.2 recognition errors per OOV-word (Adda-Decker & Lamel, 2000).

So, to get a maximal lexical coverage, the lexicon consists of two parts: the word lists on one hand and a compounding module on the other hand. Word lists are carefully chosen to be the basic building blocks for the most frequent words in Dutch. All the compounds that can be formed by the software module on the basis of simplex base words have been removed from the word lists. This procedure of decompounding is also used by Ordelman, van Hessen & de Jong (2001): A decompounding procedure could improve the lexical coverage when infrequent compounds are to be expected. It enhances the lexical coverage, as instead of having 40.000 entries in our lexicon, the lexicon can consist of 40.000 base forms, with which compounds can be formed. The creation of the word lists is described in section 2. The design of the automated compounding module is described in section 3.

To test the automated compounding module, a test program was written that produces accuracy percentages for the compounding module. This test program and the test results are described in section 4.

2. The Creation of the Word Lists

To create accurate word lists containing the building blocks with the highest lexical coverage, we started with the CGN-lexicon (Piepenbrock, 2000) and began reducing this in several steps.

The CGN-lexicon is a very large lexicon containing word forms, parts of speech, morpho-syntactic information, lemmas (Van Eynde, Zavrel, & Daelemans, 2000) and phonological information. It has about 570.000 entries.

A first reduction was the reorganization of the CGN-lexicon to a one entry per word form per part of speech (POS) lexicon.

For instance, the word *werk* (work) appeared several times in the CGN-lexicon. Once as a noun, and twice as a verb.

werk -> N(soort,ev,basis,zijd,stan)

werk -> WW(pv,imp,ev)

werk -> WW(pv,tgw,ev)

The verb-form of “werk” has a morpho-syntactic tag for its use as a singular present tense verb, and it has another morpho-syntactic tag for its imperative use. As a result of the first reduction, the word *werk* remains as two entries in the lexicon: once as a noun and once as a verb.

werk -> N(soort,ev,basis,zijd,stan)

werk -> WW((pv,imp,ev) | (pv,tgw,ev))

The verb entry has a morpho-syntactic tag that is a merger of the two syntactic verb tags of the word. This first reduction is done without information loss. About 300.000 entries remained.

A second reduction was the removal of the compounds from these 300.000 entries. Every compound that can be formed by the module described in section 3 was removed, and this resulted in about 150.000 entries. A big part of this reduction is due to the occurrence of separable verbs (approximately 50.000 entries).

A next step in the selection of the words contained by the word lists was the selection of the most frequent entries. A number of word-frequency lists for Dutch¹ were combined to get a large data set on which the frequencies are counted.

A first list is created by taking the 36.000 most frequent words. This list is called the Basic-Word-list (BWL).

For the building blocks which do not occur by themselves, a second list is created, called the Quasi-Word-list (QWL). The idea of using a separate list for building blocks that cannot exist by themselves was already tested and implemented by Verloren van Themaat (1972)! The QWL contains all the word parts from the CGN-lexicon which cannot occur by themselves, but which are variations on words that do exist by themselves.

For example: the word *zonneshijn* (sunshine) consists of two parts, the part *zonne* (sun) and the part *schijn* (shine). The part *zonne* is not a lexical entry by itself, but it corresponds to the lexical entry *zon*. The part *zonne* is the word *zon* plus a binding phoneme. The part *zonne* is added to the QWL, while the word *zon* is part of the BWL.

The QWL was constructed by taking the list of compounds that were removed from the lexicon during lexical reduction. From this list, all the modifiers that were not valid words were taken. Surprisingly, this concerns only a small amount of entries: The QWL contains about 1000 entries, because a lot of modifiers and binding phonemes form the plural of the modifier, or they form the adjective with *s* form, as the binding phoneme is often an *s*. If we take the compound *hemelsblauw* (heavenly blue), we have the word *hemels* and the word *blue*. The word *hemels* in the compound is the singular word *hemel* plus the binding phoneme *s*, but this is equal to the adjective *hemels*. These are considered existing words, and are not added to the QWL.

With the combination of the Word List and the QWL the lexicon is designed in such a way that the applicability of the rules used in the compounding module is maximized.

3. The Compounding Module

To allow for real-time lexical expansion, the sequence of words that result from the recognition process are offered as input to the compounding module. The current module allows for two up to five input base forms.

For each input unit, the module checks if this input unit is part of the BWL or of the QWL. If an input unit is found in the QWL, it gets a 'Quasi'-tag. If the input unit appears to be a word found in the BWL, the word is looked up in the CGN lexicon (Piepenbrock, 2000), and all its possible morpho-syntactic tags are retrieved. For a description of the morpho-syntactic tags, see Van Eynde (2000).

¹The frequency lists used are the CGN-frequency list from release 7.0 with 3.073.251 words, the ILK-frequency list (Induction of Linguistic Knowledge, see <http://ilk.kub.nl>) with 455.885 words, the Celex frequency list with 40.252.523 words, the De Standaard frequency list with 32.544.615 words and the MLCC frequency list (the Dutch part of the Multilingual Corpora for Cooperation, see <http://www.issco.unige.ch/projects/MLCC.html>) with 7.252.109 words. These are all frequency lists based on Dutch corpora.

For instance: if we input *zonne+schijn*, the result of the first step will be:

zonne -> QUASI
schijn -> N(soort,ev,basis,zijd,stan) | WW((pv,imp,ev) | (pv,tgw,ev))

For each of the tags of each input unit, it is checked whether the rules are applicable. If the rules fire, then the resulting word is a potential compound, with its morpho-syntactic tag. If none of the rules fire, no compounds can be formed with these units, and the input units are identified as noncompounds.

For the resulting compounds, their compounding confidence is calculated. How this is done is explained in section 3.2.2.

3.1. Declarative Knowledge: the Rules

If more than two input units are given, the module tries to compound them two by two, recursively. The recursion tries left branching first. If left branching does not succeed, right branching is tried. The branching which results in the compound does not always reflect the semantic branching, but it yields the correct compound.

An example of left branching is *samenstellingsprobleem* [E: compounding problem]: ((*samen+stellings*)+*probleem*). An example of right branching is *overgrootmoeder* [E: great grandmother]: (*over*+(*groot+moeder*)).

All the rules stated here are rules concerning two input units: the modifier and the head. As is stated in Booij & van Santen (1995) and De Wachter & Provoost (1993), the head determines the syntactic category of the compound. The compound hence inherits the tag from its head. In regular compounds, the head is always the right hand side of the compound.² A compound can be formed:

- If the head is a noun,
 - and the modifier is a quasi-word (*hersens+groot* [E: brain growth]), a preposition³ (*voor+ruit* [E: wind shield]), an adverb⁴ (*samen+stelling* [E: compound]), a noun (*spraak+herkenning* [E:

²There are some exceptions to this rule, but these cannot be formed with the compounding module. The solution we chose for such words was to put them in the BWL, as a compound. We consider those exceptions to be lexicalised. A word like *na-af* is a preposition + a noun, but the compound is a verb. See Booij & van Santen (1995) for more examples.

³Not all prepositions can be used in compounds. This is a list of the prepositions which can be modifiers: *aan, achter, af, bij, binnen, boven, buiten, heen, langs, mede, mee, na, om, onder, op, over, rond, tegen, toe, tussen, uit, voor*. This restriction applies to all the rules where prepositions are involved.

⁴Not all adverbs can be used in compounds. This is a list of the adverbs which can be modifiers: *aaneen, achterna, achterom, achterop, achterover, bijeen, dooreen, ineen, mis, neder, neer, omhoog, omlaag, omver, ondereen, onderuit, opeen, overeen, overhoop, samen, tegemoet, teloor, teniet, terug, teweeg, tewerk, thuis, uiteen, vaneen, vast, verder, vooraf, voorop, voorover, voort, vooruit, weder, weer, weg*. This restriction applies to all the rules where adverbs are involved.

speech recognition]), or a verb in its base form⁵ (*speel+plein* [E: playground]).

- and the head is non-neuter and the modifier is an adjective in its base form⁶ (*mooi+prater* [E: humbug]).
 - (If the head is neuter and singular, compounding remains possible with a base form adjective, but becomes indistinguishable from the cases where the normal word order ADJ N concerns two separate words. If we allow compounding in this case, a large number of false hits will be generated (*klein kind* [E: small child] vs. *kleinkind* [E: grandchild].)
- If the head is an adjective,
 - and the modifier is a quasi-word (*hersen+dode* [E: cerebral dead]), a preposition (*door+nat* [E: wet through]), a noun (*kurk+droog* [E: bone-dry]), a verb in its basic form (*spil+ziek* [E: spendthrift]), or an adjective in its basic form (*licht+grijs* [E: light gray]).
 - (If the modifier is an adverb, although compounding is possible, the compound is not accepted because it becomes indistinguishable from the cases where the normal word order ADV ADJ concerns two separate words. If we allow compounding in this case, a large number of false hits will be generated (*wel lustig* [E: rather cheerful] vs. *wellustig* [E: voluptuous].)
 - If the head is a past participle and the modifier is a noun, the compound is an adjective (*brug+gepensioneerd* [E: retired early]), since there is no corresponding verb (*brugpensioneren**). In that case, the head is considered as an adjective.
 - If the head is a verb,
 - and the modifier is a preposition (*mee+werken* [E: contribute]), an adverb (*samen+werken* [E: cooperate]), or an adjective in its basic form (*hoog+springen* [E: high jumping]).
 - If the head is a past participle, it can be considered as an adjective. See the rules for adjectives.
 - Although the compounding of two verbs occurs (e.g.: *schuddebollen* [E: to dodder]), it is not productive (Haeseryn et al., 1997), so it is not included in the rules.
 - If both head and modifier are quantifiers (*zeven+tien* [E: seventeen]).

⁵In the case of verbs, a base form is the stem of the verbs. In Dutch, the first person singular and the imperative singular usually equal the stem.

⁶In the case of adjectives, a base form is represented in the syntactic tag with the feature values “basis,zonder”. This means that it has no declension (-e) or degree affix (-er, -st).

These rules were derived from Booij & van Santen (1995) and Haeseryn et al. (1997). They are applied recursively when the input contains more than two items, and they tend to overgenerate. There are more possible rules than the ones mentioned here. The reason why other rules⁷ are not included here is that they generate more false hits than correct compounds.

3.2. Statistical Knowledge

3.2.1. The Relative Frequency Threshold Parameter

Because of the occurrence of homonymy between some prepositions or adverbs and some nouns, this parameter was added to the compound module. For instance, the preposition *bij* (with) is a homonym of the noun *bij* (bee). The prepositional occurrence of the word has a much higher frequency than the nominal occurrence of the word. To take this fact into account, the parameter Relative Frequency Threshold (RFT) was added to the compound module. This parameter can be set on any value between 0 and 1. The combined frequency list, which was mentioned earlier in this paper, was converted into a relative frequency list with word forms as entries. So, for each word form we get the relative frequency of a certain part of speech as opposed to the relative frequency for the same word form, but with another part of speech. If the relative frequency of a word form-POS combination doesn't exceed the RFT, this POS is not taken into account for the given word form.

Example: Set the RFT-parameter to 0.05. If we have the word *bij* as a potential modifier, there are two possible POS's for *bij*, with the $Freq(bij_{PREP}) = 4689$, and the $Freq(bij_N) = 2$, so the relative frequency of *bij* as a preposition is $RFR(bij_{PREP}) = \frac{4689}{4691} = 0.999 > RFT$. The relative frequency of *bij* as a noun is $RFR(bij_N) = \frac{2}{4691} = 0.0004 < RFT$, so the use of the word *bij* as a noun gets rejected by the RFT-parameter.

Two different implementations of the RFT parameter were tested: in one implementation, the RFT was only used on the modifiers but not on the heads; in the other implementation the RFT was used on both the modifier and the head. The latter consistently showed the best results.

3.2.2. The Compound Probability⁸

To get a confidence measure about the compound, we estimate the following ratio:

$$\frac{P(\text{compound}_{(w_1=\text{mod}_i, w_2=\text{head}_j)})}{P(\text{compound}_{(w_1=*, w_2=\text{head}_j)})}$$

where

$P(\text{compound}_{(w_1=\text{mod}_i, w_2=\text{head}_j)})$ is the probability that two consecutive words (members of BWL or QWL) form a compound, rather than being two separate words, and

$P(\text{compound}_{(w_1=*, w_2=\text{head}_j)})$ is the probability of the second word part as a head, with any modifier.

- If the compound is found in the frequency list the ratio is estimated this way:

⁷E.g.: Noun + Verb

⁸I would like to thank Kris Demuyne of ESAT for helping me with the statistical reasoning described in this section

$$\frac{P(\text{compound}_{(w_1=\text{mod}_i, w_2=\text{head}_j)})}{P(\text{compound}_{(w_1=*, w_2=\text{head}_j)})} \approx \frac{\text{Freq}(\text{compound}_{(w_1=\text{mod}_i, w_2=\text{head}_j)})}{\text{Freq}(\text{compound}_{(w_1=*, w_2=\text{head}_j)})} \times (1 - D_{\text{head}_j})$$

where

$\text{Freq}(\text{compound}_{(w_1=\text{mod}_i, w_2=\text{head}_j)})$ is the frequency of the compound that consists of w_1 as modifier and w_2 as head, and

$\text{Freq}(\text{compound}_{(w_1=*, w_2=\text{head}_j)})$ is the frequency of the second word part as a head, with any modifier; it is the sum of the frequencies of all the words with the second word part as a head.

D_{head_j} is the amount of the total probability which is reserved for words that cannot be found in the frequency list. D_{head_j} is called the discount parameter, and is estimated like this:

$$D_{\text{head}_j} = \frac{\#\text{different}(\text{mod}|\text{head}_j)}{\text{Freq}(\text{compound}_{(w_1=*, w_2=\text{head}_j)})}$$

where

$\#\text{different}(\text{mod}|\text{head}_j)$ is the number of different modifiers occurring with the given head,

$\text{Freq}(\text{compound}_{(w_1=*, w_2=\text{head}_j)})$ is the frequency of the second word part as a head, with any modifier; it is the sum of the frequencies of all the words with the second word part as a head.

D_{head_j} is the discount from the total frequency for the unseen items. For more information on this backing-off procedure, see Placeway et al. (1993),

$(1 - D_{\text{head}_j})$ is the amount of the total probability which is reserved for words that can be found in the frequency list.

If we want to calculate the ratio for the compound *binnenkijken* (look inside), which consists of two parts: *binnen* [E: inside] and *kijken* [E: look]. The compound *binnenkijken* occurs 10 times in the corpus. The head *kijken* occurs 2188 times as a head in a compound, with 21 different modifiers. If we fill in these numbers in the formulas we get: $\frac{10}{2188} \times (1 - \frac{21}{2188}) = 0.0045$.

- If the compound is not found in the frequency list, the ratio is estimated this way:

$$\frac{P(\text{compound}_{(w_1=\text{mod}_i, w_2=\text{head}_j)})}{P(\text{compound}_{(w_1=*, w_2=\text{head}_j)})} \approx D_{\text{head}_j} \times \frac{\text{Freq}(\text{compound}_{(w_1=\text{mod}_i, w_2=*)})}{\text{Freq}(*)}$$

where

$\text{Freq}(\text{compound}_{(w_1=\text{mod}_i, w_2=*)})$ is the frequency of the first word part as a modifier, with any head; it is the sum of the frequencies of all the words with the first word part as a modifier,

$\text{Freq}(*)$ is the total frequency of all words, it is the sum of all frequencies of all words.

If we want to calculate the ratio for the compound *frequentietabel* (frequency table), which consists of two parts:

frequentie [E: frequency] and *tabel* [E: table]. The compound *frequentietabel* does not occur in the corpus. The head *tabel* occurs 141 times as a head in a compound, with 17 different modifiers. The modifier *frequentie* occurs 15 times as a modifier in a compound. The total sum of all frequencies in the frequency list is 79.862.581. If we fill in these numbers in the formulas we get: $\frac{17}{141} \times \frac{15}{79862581} = 2.26e^{-8}$

The use of these formulas ensures us that the resulting value will always be between 0 and 1. The higher it is, the more probable the compound becomes.

As will become clear from the test results, this value gives us a great hand in estimating if the compound is real or not. It helps us in finding out whether the rules are over-generating or not.

4. Testing the Module

In the first part of this section, you will find a description of the test system. In a second part of this section, you will find some test results. In the third part, test conclusions are drawn.

4.1. The Test System

The test system takes a text as its input. It converts this text into a list of words. All punctuation marks are converted into “#” symbols. On a first run, every word in the list is looked up in the BWL. If the word is not present in the BWL, it is checked whether the word can be automatically split into two parts, a modifier and a head, for which the modifier either is present in the QWL or in the BWL, and the head is present in the BWL. If this is the case, the test system supposes that the word is a compound with two parts. If the split up procedure didn’t result in a two-part “compound”, the system tries to split the word into three parts, of which the first and the second part should be found in the QWL or in the BWL, and the third part should be found in the BWL. In the splitting up procedure, the rules are not used, in order to test the accuracy of the rules independently from the input they get.

If a word is not found in the BWL, and seems impossible to split up into either two or three parts, the word is classified as an out-of-vocabulary word. For all the other words, we know they consist of one, two, or three parts. But we want more than the percentage of OOV-words. We want to know the accuracy of the compounding module. The test system tests this accuracy for compounds with two parts and for compounds with three parts. Compounds with four or more parts are classified as OOV-words, but as this concerns a very limited amount of words, it hardly affects the results. Note that these words can be compounded with the compounding module, as the module allows up to five word parts. The restriction of maximum three parts is a restriction of the test system, and not of the compounding module.

The test system takes a second run through the BWL, and now uses the units that were identified as word parts in the first run. Every two consecutive parts are sent to the compound module, even if they belong to different words,

because we also want to test how much the compound module overgenerates. The compound module has three possible return values:

- The first word part cannot be compounded with the second. It is identified as a noncompound word. If the second part is a punctuation mark, the first word is always identified as a noncompound.
- The first word can be compounded with the second, according to the rules, and the compound word is confirmed by the CGN-lexicon (full version).
- The first word can be compounded with the second, according to the rules, but the compound is not confirmed by the CGN-lexicon (full version).

We make a distinction between compounds confirmed by the full CGN-lexicon, and compounds which are not confirmed, to get a detailed analysis of the accuracy of the rules and to get a clear view on what kind of compounds are found or missed with the compounding module. As the test results in section 4.2.3 show, this distinction is useful.

The returned result from the compound module is then compared with the result for the same word in the first run. In table 1, the values of the outcome of the first and second run and their resulting values are presented.

First Run	Second Run	Results
NC	NC	correct NC id.
NC	CGN	false hit CGN id.
NC	UNC	false hit UNC id.
COMP	NC	false hit NC id.
COMP	UNC	correct UNC id.

Table 1: First and Second Run Outcome and Resulting Values (NC=Noncompound,COMP=Compound,CGN=CGN-confirmed,UNC=unconfirmed)

If the word is identified as a three part compound in the first run, the compounding results for the first two parts, and for the second and the third part are irrelevant⁹. At this point we get results on the accuracy of the compounding module for two-part compounding. We also want to know the accuracy for three-part compounding. Analogous to the description of the two-part compounding, every three units are sent to the compounding module. The possible outcomes are the same as for the two-part compounding. Results are compared with what the first run taught us about the input, and the accuracy of the compounding module for three-part compounding can be calculated.

As every two and three consecutive word parts are tested, this is an under-estimation of the performance the system would have in a real recognition system, as word

⁹An example: For the input word *overgrootmoeder* (E: *great grandmother*) it is irrelevant if “over+groot” can form a compound, as the input text does not contain the word “overgroot” nor the words “over” and “groot”, hence we cannot score the correctness of compounding “over+groot”.

parts which are already used in compounding with the previous word part, are tried again to compound with the next word part.

In the test system, as it is used for generating the test results, described in section 4.2, a BWL of approximately 35.000 words and a QWL of approximately 1000 words was used. This means that we are not yet at the limits of what can be put in the speech recognizer lexicon (approx. 40.000 entries).

4.2. The Test Results

4.2.1. The Test Texts and their Properties.

Three texts were used as input texts. The texts differ in language use. Here is a description of the input texts:

- *Thuis.txt* is one episode of a Dutch soap series. Only the dialogues were used. The language used in this text is rather simple, on the edge of dialect. The amount of compounds in this text is limited (50 words). The text contains 3415 words.
- *Aspe.txt* is the first chapter of a novel, written by Pieter Aspe. The language used is very literate, with a high amount of compounds (279 words). The text contains 4589 words.
- *Interview.txt* is an interview that was transcribed for the CGN-project. The language used is spontaneous speech. The amount of compounds is in between the two other test texts (131 words). The text contains 4645 words.

To get an idea about the complexity of the test texts, and the amount of compounds occurring in those texts, a compound counter was added to the test program: when the test program is performing its first run through the input text, each word gets a code for whether it is a noncompound, a 2-part compound, a 3-part compound or an OOV-word. The frequency of these codes is counted. The result is the relative frequency of noncompounds, 2-part compounds, 3-part compounds, and OOV-words on the total amount of words, as shown in table 2.

Text	NonComps	2-parts	3-parts	OOV
Thuis	95.45%	1.41%	0.06%	3.08%
Aspe	90.15 %	5.64%	0.44%	3.77%
Interview	96.21%	2.84%	0.11%	0.84%

Table 2: Test Text Complexity

For the three input texts, the amount of OOV-words is rather small. Most of the OOV words are proper nouns. In *Thuis.txt* we find a high amount of words ending with the diminutive *-ke*, which is not considered standard Dutch. In *Aspe.txt* there are a number of “dialogues” with English, French, or dialect sentences, which generate a high amount of OOV-words. In the *Interview.txt*, the amount of OOV-words is very small.

4.2.2. The Correct Identification of Noncompounds and Compounds

The amount of correct identification is dependent on the threshold values of the parameters¹⁰. For the RFT parameter, we show two values: RFT=0 and RFT=0.05, as the results didn't vary significantly on other values of the parameter.

For the Compound probability different values were tried in order to get a maximum recognition accuracy. Figure 1 shows how many of the input words are correctly identified by the compound module as being a noncompound, a 2-part compound or a 3-part compound. Note that a word is identified as a noncompound only if it is not a 2-part compound and not a 3-part compound. These two conditions need to be true.

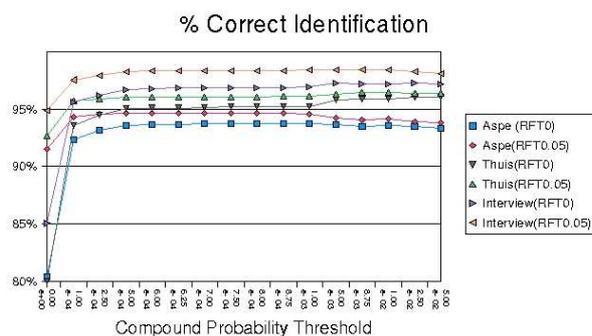


Figure 1: Correct Identification Rate

Table 3 shows at which parameter value the highest recognition rate was reached.

Text	RFT	CPT	% Correct Id
Aspe	0.05	0.003	94.53 %
Thuis	0.05	0.003	96.28 %
Interview	0.05	0.003	98.47 %

Table 3: Highest Recognition Rates

The most salient facts in figure 1 is the strong rise in correct identification as soon as the RFT-parameter is used. The introduction of the RFT-parameter shows to be a good idea.

Another strong rise in correct identification percentage can be noted at the introduction of the Compound Probability Threshold. If the value of the compound probability parameter of the compound (as described in section 3.2) is below the threshold set by the test system, the compound is not accepted. The values listed in the CPT column of table 2 are the average values of the ranges where the highest identification percentage was reached.

These values show a nearly perfect negative correlation of -0.98 with the amount of compounds in the test text. The more compounds present in the text, the lower the parameter threshold should be set to reach a maximum identification percentage.

¹⁰Parameters are described in section 3.2

4.2.3. Accuracy of the Compound Module

In figure 2, you can see the results split out over different conditions. This enables us to spot the strong and weak points of the module.

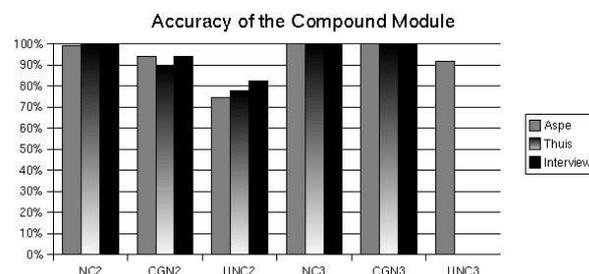


Figure 2: Accuracy of the Compound Module

The strong points in the system are clearly the identification of noncompounds for two and three part compounding, and the identification of 3-part CGN confirmed compounds. The identification of 2-part CGN confirmed compounds ranges between 89% and 94.5%. The identification of 2-part unconfirmed compounds ranges between 74% and 82.5%. Although this is much lower than in most other conditions, it should be noted that we did some tests before the Compound Probability Threshold was used, and these results were at 19% accuracy maximum. The biggest gain by introducing the CPT was for 2-part unconfirmed words.

The identification of 3-part unconfirmed compounds is 91.5% for Aspe.txt, and 0% for the other two texts. For Aspe.txt, before the CPT was used, a maximum of 23% could be reached for this value. The 0% results are due to the fact that there are very little unconfirmed 3-part compounds in Thuis.txt and in Interview.txt. Even if we could get these words to be correctly identified as unconfirmed 3-part compounds, this would have little or no impact on the total correct identification percentage.

4.3. The Test Conclusions

The test results show that the general identification of words as consisting of one, two, or three parts is between 94.5 and 98.5%, if the RFT-parameter is used and the CPT-parameter is optimized.

It should be noted that the OOV-rates are rather small, considering that the real lexicon size used for these tests is 35.000 items in the BWL and 1.000 items in the QWL.

The results show that the compounding module is performing extremely well on the identification of noncompounds: if the module says no compounding is possible, you can be sure that this is true. If the module, on the other hand, tells you that compounding is possible (be it with two or three parts), the Compound Probability gives a useful cue as to whether compounding is also plausible. Depending on the text complexity, the value can help in the decision to continue the recognition path with a compound word or with separate words. Some mistakes the module can make are due to the fact that some word parts can be written together and apart, having a different meaning, depending on

the spelling¹¹.

5. General Conclusions

The building of a lexicon that can be extended in real-time, to capture some of the productive processes which occur in Dutch proves to be successful. The building of a compounding module that generates information about the *compoundability* of the input word parts proves to provide a way of reaching a high lexical coverage with a limited lexicon size.

As the lexicon can be expanded in real time, by combining word parts using the compound module, the entries present in this expanded lexicon are practically limitless. The number of entries which is present in the BWL and the QWL can be kept under 40.000, without compromising the lexical coverage.

The aim of building a lexicon that comprises already existing compounds proves to be successful. The aim of building a lexicon that comprises new words based on already existing word parts seems a lot harder. New words are included in the lexicon, through the real-time expansion module, but the rules are only applicable with a low accuracy. By using frequency information based on a large data set, the accuracy can be lifted to a reasonable amount.

We can conclude by saying that designing a lexicon with a minimized out-of-vocabulary rate and a limited size can be done in a successful way by real-time lexical expansion through automated compounding.

6. Acknowledgments

Research funded by IWT (Institute for Innovation in Science and Technology) in the STWW program, project ATraNoS (Automatic Transcription and Normalisation of Speech). The research consortium consists of CCL and ESAT (K.U.Leuven), CNTS (U. Antwerpen), and ELIS (U. Gent). The project aims at generic basic research on speech recognition and normalization of unrestricted speech with automatic subtitling from speech as a case study.

7. References

- Adda-Decker, M. & Lamel, L. (2000). The use of lexica in automatic speech recognition. In F.Van Eynde & D.Gibbon (Eds.), *Lexicon Development for Speech and Language Processing*, pages 43–75, Kluwer Academic Publishers. Dordrecht.
- Booij, G. and van Santen, A. (1995). *Morfologie. De woordstructuur van het Nederlands*. Amsterdam University Press, Amsterdam.
- De Wachter, L. & Provoost, J. (1993). A computational interpretation of compounds. In F.Van Eynde and D.Heylen (Eds.), *Working Papers in Natural Language Processing*.

Haeseryn, W., Geerts, G., de Rooij, J. & van den Toorn, M. (1997). *Algemene Nederlandse Spraakkunst*. Martinus Nijhoff Uitgevers, Groningen.

Ordelman, R., van Hessen, A. & de Jong, F. (2001). Lexicon optimization for Dutch speech recognition in spoken document retrieval. *Eurospeech 2001*.

Piepenbrock, R. (2000). CGN Lexicon 7.0. Corpus Gesproken Nederlands.

Placeway, P., Schwartz, R., Fung, P. & Nguyen, L. (1993). The estimation of powerful language models from small and large corpora. *Proceedings of the ICASSP 1993*, Vol.2, pp. 33-36, ICASSP, Minneapolis, Minnesota, USA.

Van Eynde, F. (2000). *Part of Speech Tagging en Lemmatisering* CCL, Leuven. Online: <http://lands.let.kun.nl/cgn/publicat.htm>.

Van Eynde, F., Zavrel, J. & Daelemans, W. (2000). Part of speech tagging and lemmatisation for the spoken dutch corpus. *Proceedings of the Second International Conference on Language Resources and Evaluation*.

Verloren Van Themaat (1972). *Automatic Analysis of Dutch Compound Words*. Vol. 38 of *Mathematical Centre Tracts*, Mathematisch Centrum, Amsterdam.

¹¹Consider the construction “binnen kijken” [E: watch indoors] vs. the construction “binnenkijken” [E: look inside] in the sentences: *Binnen kijken we televisie*. [E: Indoors we watch television.] vs. *Als men niet mag binnenkijken, moet je de gordijnen sluiten*. [E: If people are not allowed to look inside, you have to close the curtains.]