Unsupervised learning of morphology in the USSR

Franck Burlot, François Yvon
LIMSI, CNRS, Université Paris-Saclay, Orsay, France

Abstract

This article deals with an important task for the processing of morphologically rich languages. Unsupervised learning of morphology mainly consists of learning a grammar that enables word segmentation into morphemes without any prior knowledge of the analysed language. It is usually assumed that the origins of such a task date back to the times of Zellig Harris, an assumption which ignores the important contribution of his contemporary, the Soviet linguist Nikolaj Dmitrievič Andreev, who developed a statistico-combinatorial model to learn morphology in the 1960s. We propose a critical description of Andreev's model and attempt to bring to light its pioneering aspects as well as its weaknesses. Finally, we show results over several European languages. Our implementation of the model can be downloaded from `https://github.com/franckbrl/stat_comb_model`.

Key words: Morphology, Soviet Linguistics, Information Theory, word segmentation, unsupervised learning, Nikolaj Andreev, statistico-combinatorial model.

1. Introduction

The task of unsupervised learning of morphology generally consists of learning a model that segments words into smaller units: morphemes[1]. It assumes as input a text segmented into words and produces a model that is similar to a grammar learned without prior linguistic knowledge, according to which the words of the same text are split into grammatically relevant morphological units. It is usually assumed that the origins of such an important task date back to the times of Zellig Harris, an assumption which ignores the important contribution of his contemporary, the Soviet linguist Nikolaj Dmitrievič Andreev (1920-1997), who spent his career at the Institute of Linguistics of the Academy of Sciences in Leningrad. In the 1960s, he has developed the statistico-combinatorial model to learn morphology in an unsupervised way, mainly described in two books: (Andreev, 1965) and (Andreev, 1967).

This model, making Andreev a pioneer in the domain, has nevertheless remained unknown internationally, and the first citations of his works in the literature are rather recent. Firstly, his work on the learning of morphology has never been translated from Russian into any other language. Secondly, the model is described in his work in a rather fragmentary manner. Indeed, when reading Andreev's descriptions of the algorithm, the impression is that it was never fully implemented and those descriptions are different pieces that we sometimes had a hard time trying to put together in our implementation. These difficulties, noticed earlier in (Hammarström and Borin, 2011)[2] and (Goldsmith, 2001), seem to be a serious obstacle to the success Andreev's work deserves. Besides, there exists, to our knowledge, only one attempt to implement his model, by Cromm (1997). Unfortunately, this short paper does not give a better understanding of the statistico-combinatorial model than the original books. We argue in

[1] The Morpho Challenge describes this task as "discover[ing] which morphemes (smallest individually meaningful units of language) words consist of" (`http://research.ics.aalto.fi/events/morphochallenge/`).

[2] "The papers describing these experiments are short, and it is not always clear exactly what has been done."

this paper that such difficulties are not a good enough reason for taking away from Andreev the important place he should be holding in the history of the task. For the sake of clarity, this article also presents our implementation of the model, that can be downloaded from `https://github.com/franckbrl/stat_comb_model`.

We first describe Andreev's statistico-combinatorial model and comment on our attempt to implement it. We then report experimental results obtained with this implementation. Finally, we put Andreev's findings into perspective by bringing their pioneering aspects to light and by investigating what is still used nowadays in state-of-the-art systems.

2. The statistico-combinatorial model

The first step of Andreev's algorithm consists of identifying the *bootstrap affix* (see 2.2.), which is used to identify stems that can associated to it. The comparison of the different affixes seen with these stems enables the identification of the first *class*, grouping together stems sharing a set of affixes (see Table 1). By repeating these steps, we obtain as output: 1) a model that segments the words from the corpus into stems and affixes; 2) a categorization of each analysed token (that shall belong to a class).

	What we look for	**Input**	**Function**	**Output**
	`type d typing gla d note d noting for meeting note fe d`			
1:	Informants (by position)	-1: `d, g, e, r` -2: `e, n, a, o, t` -3: `p, i, l, t...`	Correlative function (§ 2.1.)	(`d`, -1), (`n`, -2), (`i`, -3), (`e`, -1), (`e`, -2)
2:	Bootstrap affix (from 1st informant)	`ed, ad`	Gradient rate (§ 2.2.)	Affix extended to `ed`
	`typ ed typing glad not ed noting for meeting note f ed`			
3:	Class stem set	Tokens	Get type stems	`typ, not, f`
	`typed typ ing glad noted not ing f or meeting not e fed`			
4:	Tokens containing class stems	Tokens	Get word forms	`typing, noting for, note`
5:	2nd affix	`ing or e`	Correlative function (§ 2.3.)	2nd affix: `ing`
	`typ ed typ ing glad not ed not ing for meet ing note f ed`			
6:	Bootstrap affix stems (M_1)	Tokens	Get type stems	`typ not f`
7:	2nd affix stems (M_2)	Tokens	Get type stems	`typ not meet`
8:	Add affixes to class	M_1, M_2	Reduction rate (§ 2.3.)	Added `ed, ing`
9:	New class stem set		Class stem set $\cap M_1 \cap M_2$	`typ, not`
10:	Go to 4: with `ing` as bootstrap affix and the new class stem set (`e` is the second affix selected in 5:).			
11:	Go to 1: and select the next informant from the list (`n` at position -2).			

Table 1: Accepting the first two affixes of the class (on a toy corpus of 9 tokens) using as first informant `d` at position -1. See text for details.

The algorithm starts by collecting a few statistics over the corpus. We obtain the unigram character probability $p(char)$ and the average word length, which is used to formulate two assumptions: 1) words shorter than $\frac{2}{3}$ of the average length are ignored and remain unsegmented;

2) the hypothetical affix length does not exceed the average word length. Within this affix length limit, the probability of characters conditioned on their absolute position in the word $p(char|pos)$[3] can be computed. Once this is done, it is possible to make more specific assumptions about the words in the corpus. The examples illustrating this description are obtained with our own implementation over the first 40,000 sentences of the News Crawl 2014 corpus.[4]

2.1. Informants

In (Andreev, 1967), the term *informant* refers to characters at a certain position in the words that are either potential affixes by themselves (such as `s` at the end of a word), or need to be extended with additional characters in order to form an affix (e.g. `g` is extended to `ing`). To obtain such informants, characters in the hypothetical affix zone of the words need to be filtered by their conditional probability. At each position in the word, Andreev keeps the characters for which the conditional probability is higher than half the maximum conditional probability for that position. For each character remaining at a given position, the next step computes what the author calls the *correlative function* (CF),[5] which measures the degree of dependency of the character with respect to its position in the word (Equation (1)).

rank	char	$p(char)$	pos	$p(char\|pos)$	CF($char$)
1	**w**	0.02	1	0.06	**4.17**
4	**d**	0.04	-1	0.11	**2.76**
8	**i**	0.07	-3	0.18	**2.38**
9	**s**	0.07	-1	0.15	**2.21**

Table 2: Characters ranked by their correlative function.

$$\mathrm{CF}(\mathrm{char}) = \frac{p(\mathrm{char}|\mathrm{pos})}{p(\mathrm{char})} \qquad (1)$$

The character that has the highest correlative function becomes the first *informant*, thanks to which we obtain the bootstrap affix. In our experiments with English (Table 2), the first three informants do not seem likely to be part of any affix and we have to wait for the fourth (`d` at position -1), which may very well extend to the suffix forming the verbal past tense `ed` (as in `segment-`**`ed`**). The progressive ending `ing` comes into right with the eighth informant `i` at position -3, followed by the third person (or plural) suffix: `s` at word end (position -1).

2.2. The bootstrap affix

The goal of the next step is to decide whether the informant already forms an affix (such as the 9th informant `s` in Table 2), or if the character needs to be extended, considering that at this point in time we have no reason to consider that `ed` (in `segment-ed`) is a better suffix than `nted` (in `segme-nted`).

In (Andreev, 1967), the author constrains the affix extension by setting a maximum affix length of $\lceil L^2/S \rceil$ (where L is the average word length and S the average sentence length), although he does not explain why the affix length should depend on those values. According to this formula, the maximum affix length in our English experiment is 1, which would be problematic.[6] No

[3] The position index (pos) starts at 1 from the beginning of the word (prefix) and at -1 from its end (suffix).

[4] `http://www.statmt.org/wmt15/translation-task.html`

[5] The similarities between the correlative function and the pointwise mutual information are discussed in 4.2.

[6] First, such a constraint would prevent extending `-d` into `-ed`. Second, the author does not specify what happens when informant `i` is located at position -3 (see Table 2), which is two characters above the maximum affix length (the suffix found from this informant cannot be shorter than three characters).

such constraint appears in (Andreeva, 1965) and we set this limit to 4 in our implementation. In order to find the right extension of the affix, Andreev (1967) resorts to what he calls the *gradient rate*. The idea is to compare the frequencies ($p_{i=1}$ and $p_{i=2}$) of the two most frequent characters near the affix in construction. Therefore the gradient rate of a given affix GR(aff) is:

$$\text{GR}(\text{aff}) = \frac{P_1}{P_2};\ \text{where}\ \begin{cases} P_i = p(\text{char} = c, \text{pos} = x+1 | \text{aff} = a, \text{pos} = x)\ \text{(Extend rightwards)} \\ P_i = p(\text{char} = c, \text{pos} = x-1 | \text{aff} = a, \text{pos} = x)\ \text{(Extend leftwards)} \end{cases}$$

Note that Baklušin (1965) describes this measure as *negentropy*. Indeed, as we shall mention in 4.1., the gradient rate has a lot in common with entropy, since it estimates the degree of predictability of the closest character to the informant. Andreev compares this measure to a threshold, which he sets to 1.5 (again without any explanation):

- GR(aff) > 1.5: the most frequent neighbour of the informant is added to the starting affix in construction and we can iterate the procedure.
- GR(aff) < 1.5: we stop the affix extension.

If the informant is at position 1 or -1, the extension is directed towards the inside of the word. Otherwise, the affix is extended towards the end or the beginning. If, at this last step, the gradient rate rejects all extensions, the informant is rejected. GR values obtained in our experiments for the informant *i* at position -3 are in Table 3 and help identify the bootstrap affix `ing`.

c_1, c_2	pos	P_1	P_2	GR	Result
n, t	-2	0.33	0.12	$2.69 > 1.5$	*n* accepted: *in*
g, e	-1	0.81	0.07	$12.48 > 1.5$	*g* accepted: *ing*
t, d	-4	0.14	0.11	$1.25 < 1.5$	*t* refused

Table 3: Extension of the informant `i` at position -3 (c_1 and c_2 are its most frequent neighbours).

$$\text{CF}(\text{aff}) = \frac{p(\text{aff}|\text{pos})}{\prod_{\text{char} \in \text{aff}} p(\text{char})} \quad (2)$$

2.3. The search for paradigms

Once the bootstrap affix has been found, we identify all the stems that were seen with it in the corpus. The resulting set of stems is smoothed by removing the most infrequent elements. The set affixes associated with these stems in the corpus are similarly computed, and smoothed. Note that this initial set of stems (see the *class stem set* of step 3 in Table 1) will be used until the end of the search for the paradigm and reduced each time a new affix is accepted (step 9).

Candidate affixes are sorted by their *correlative function*, which is computed as the relation of the conditional probability of the affix to its marginal probability. The conditional probability of the affix is obtained as the count of words that contain it at the corresponding position (*pos* being either the beginning or the end of the word), normalized by the vocabulary size. The marginal probability is seen by the author as the random co-occurrence of the characters it is composed of, computed as the product of the unigram probabilities of each character in the affix (Equation (2)). In our experiments, the bootstrap affix `ing` first helps the identification of the candidate affix `ed`, which has a conditional probability of 0.06 and a marginal probability of 0.004, giving the highest CF (11.82). Those two affixes can enter the paradigm, provided that they come under inflection (as opposed to derivation). The author's strategy to implement such a distinction relies on the observation that the number of stem types observed with one affix

is not too different from the number of stem types seen with the second one, if both express inflection.[7] Besides, Andreev specifies that this difference should take into consideration the corpus size in order to address data sparsity issues. He introduces a coefficient K that should allow the relation of both quantities to be better adapted to a specific corpus. Without further explanation, he recommends Equation (3), obtained after several experiments that he does not describe (L is the average word length and V the vocabulary).

$$\log_{10} K = \frac{\log_{10} L}{1 + 0.02 \log_{10} V} \quad (3) \qquad R = \frac{|M_1| - |M_2|}{K \times |M_1|} \quad (4)$$

This *reduction coefficient K* is used to compute the *reduction rate R* of both affixes, based on the length of the sets of stems seen in the corpus with the bootstrap affix (M_1) and with the second candidate affix (M_2, see Equation (4)). R is compared to the *reduction threshold* T, which is set to $1/L$ (without explanation), which leads to two situations:

- $R > T$**:** The second affix does not become part of the paradigm. Step 5 (see Table 1) is repeated with the next second candidate affix identified by the bootstrap affix;[8]
- $R < T$**:** Both affixes are added to the paradigm. The second candidate affix becomes the bootstrap affix, according to which we search for the new second affix. In this case, this next step is performed only with the stems that are common to both accepted affixes and the class stems (step 9 in Table 1).

These steps are repeated until one of three situations occurs: 1) there are only two stems left; 2) there are no more second candidate affixes; 3) if several affixes in a row have a gradient rate above the threshold. We added an extra condition for accepting a new affix: the class stem set must contain at least $\frac{1}{10}$ elements from the previous remainder set, right before the new affix is accepted. This constraint should ensure a certain consistency of the type.

Candidates	M_1	M_2	Reduction	Result	Class stems
ing, ed	636	828	$0.06 < 0.23$	*ing, ed* accepted	342
ed, es	828	589	$0.08 < 0.23$	*es* accepted	85
es, e	589	1552	$0.16 < 0.23$	*e* accepted	73

Table 4: Reduction rate for the affixes obtained using `ing` *as a bootstrap affix. The last column indicates the number of class stems that are common to both affixes.*

After this procedure, we obtain a class composed of a set of affixes corresponding to a set of stems. Note that the latter is often rather small, since each time an affix is accepted, we only keep the stems that appear with this affix and the bootstrap affix.

Andreev does not reveal how such classes are used to tag and segment the corpus. Indeed, sticking to the stems that belong to one class might result in a very low recall. On the other hand, segmenting all the words containing an affix from a class might hurt precision. The class in Table 4 obviously corresponding to verbs, if we segment all words ending in `e`, not only will we tag nouns as verbs, but we will also segment words that should not be split (e.g. `tre-e`).

Once the first class is obtained, we can process the next informant character. We finally repeat

[7] In our English corpus, we observe 435 different stems with the prefix *pro*, against 1647 with *co*. On the other hand, *ing* and *ed* correspond respectively to 2936 and 3620 different stems, showing closer figures for inflection.

[8] If this happens with the first pair of candidate affixes for the class, no class is created, affixes are rejected and the algorithm resumes with the next informant character (step 1 in Table 1).

those steps until all the informants have been processed. Note that the algorithm we describe does not take into account more than one affix in a row and thus ignores situations like the Russian verb sad-iš-sja. Although Andreev (1967) claims to address complex cases such as agglutination (p.42), he describes the procedure in rather fuzzy terms, and we were not able to implement this function. The author also mentions a way to split certain types into sub-types (p.49), and finally claims to process Russian palatalization, where a consonant varies with the suffix, as in gljadet' - gljažu. Unfortunately, his descriptions of those procedures are very short and unclear (p.49).

Cromm (1997) introduces Andreev's model as an affix identifier. The statistico-combinatorial model is in fact a lot more ambitious than this and Andreev claims to perform a word clustering task by assigning a type to each word in the corpus, while performing in parallel a segmentation task. Andreev (1967) finally describes a way to add to existing classes words that were so far undefined, by exploring the context of the words already belonging to a class. However, from this part of the algorithm on, the author stops giving empirical results and we have no reason to believe that he actually ever implemented it.

3. Experimental results

We provide experimental results of Andreev's model for English, French and Russian. In those experiments, we consider only suffixes, since we know that inflection only concerns word endings. Without this constraint, a few prefixes would be accepted and lead for example to a class composed of pro and the null prefix applied to stems such as tested, grams, vision, etc. We also set the minimum number of stems to 10, as opposed to 2 suggested by Andreev, because we observed that the last accepted affixes were often inconsistent with the class.

Class	Affixes	Stems	Examples
1	ing, ed, es, e, er	15	vot, lov, provid
2	ies, y	287	all, abilit, dut
3	s, *null*	110	attempt, fight

Table 5: Classes for English.

Class	Affixes	Stems	Examples
1	es, e, s, *null*	1074	adéquat, retenu
2	s, *null*	445	genre, livre, coffre
3	ions, ion, ive, if, eur	16	récept, oppress, agress
4	aire, és, ées, er, ée	10	honor, fragment, not
5	rait, ons, ant, ent, re, ait, ez	12	remett, perd, mett

Table 6: Classes for French.

The English classes reported in Table 5 account for the two plural forms (2, 3) and the regular verb conjugation for stems ending in e (1). The model adds the agentive -er to this class, even though is not a verbal inflection. In the process of the third type construction, affixes t and ts were accepted, but we applied one of Andreev's rules stating that if all the suffixes of the class start with the same letter (t here), this common letter should be transfered to the stem (tex-ts, text-s). Therefore the stems in this class all end with the letter t.

For the French experiments (Table 6), we used 1M sentences from the Europarl corpus (Koehn, 2005). The model found the full paradigm of French regular adjectives (1) and nouns (2), although the latter was obtained through the validation of affixes re and res for which the first two characters were transfered to the stem. (3) contains derivational endings, although we notice that inflection distinguishes both ions from ion (plural and singular) and ive from if (feminine and masculine with a consonant shift between v and f). (4) gathers a few verbal endings, but did not go far during the search, mainly because the first bootstrap affix -aire

actually comes under noun derivation. (5) is a much better verbal paradigm (verbs with an infinitive in `re`), despite its sparsity.

Class	Affixes	Stems	Examples
1	oj, yx, ogo, ymi, yj, uju, omu, ym, ye, aja, om, oe, o, y, a	40	ser'ezn, zdorov, osob
2	t', lsja, tsja, li, la, lo, l	54	ustroi, otpravi, vloži
3	enija, it', ili, at', ajut, aet, ali, ala, al, ila, il, eny, ena, eno, en	10	soverš, razreš, poluč
4	l'nyx, ej, jam, ', em, ja	10	nabljudatel, stroitel, zritel
5	enie, it', ili, at', ajut, aet, ali, ala, al, ila, il, eny, ena, eno, en	10	primen, rasšip, ustran
6	osti, ymi, yx, ogo	10	ser'ezn, slab, častn
12	ovat', ujut, uet, ami, ax, am	10	interes, atak, miting

Table 7: A few classes for Russian.

The algorithm created 14 classes for Russian (see Table 7) over 1M sentences from the News Crawl 2015 corpus. The first type contains the whole paradigm for Russian adjective (long and short forms), except the masculine short form (null affix) which is harder to retrieve due to the insertion of a mobile vowel (e.g. `opasnyj` - `opas`**`e`**`n`). Note that this exact affix set was found twice (i.e. both times starting from different informants), but unfortunately the second time did not bring any new stems to the type. We further obtain three verbal types (2, 3, 5), the latter being the most complete, containing also passive forms, although the search started from a noun derivative affix (`enie`). Finally, (12) starts with three verbal endings, then accepts three noun affixes: `ami, ax, am` are the only noun endings we got in these experimental conditions.

Our implementation of Andreev's model retrieved many affixes and most of them are correctly segmented. The reduction rate employed by Andreev to distinguish inflection and derivation works well, considering the simplicity of the assumption behind it (among the 9 affixes retrieved for English, only one is derivational). The main default we observe in these experiments is the sparsity of the classes: one gold paradigm can be divided into several sparse classes (e.g. Russian classes 2, 3, 5). Nevertheless, those classes keep a certain degree of inner consistency.

4. Putting the task into perspective

Most of the literature on the subject[9], considers that the history of unsupervised learning of morphology starts with Zellig Harris, who in (Harris, 1955) [10] laid the foundations of principles that were used afterwards. In this closing section, we try to identify the common aspects of these two models, focusing on an issue adressed by both both Harris and Andreev: morpheme identification.

4.1. Using entropy to spot morpheme boundaries

Harris sees the question of morphological analysis as the identification of morpheme boundaries based on *successor variety*. Given a corpus, a word α is considered from the first character up to the character at position i ($1 \leq i \leq$ word length). The successor variety of α_i corresponds to the number of different characters seen at position $i+1$ in the words starting from the sequence α_i.[11] The assumption is that a higher variety marks a morpheme boundary. In order to segment a word, Harris proposes different strategies:

[9] See (Goldsmith, 2001, 2010; Hammarström and Borin, 2011).
[10] Therefore, eight years before the first article on the statistico-combinatorial model in (Andreeva, 1963).
[11] The same method is applied in the other direction, using the *predecessor variety*.

- he manually sets a threshold K and segments whenever the successor (or the predecessor, or both) variety exceed it (Harris, 1955).
- he segments whenever the successor at position $i+1$ has a higher or equal variety to the one at position i. This strategy consists in finding *relative peaks* in the varieties and dispenses with the delicate choice of absolute thresholds (Harris, 1967).

The main disadvantage of using successor variety is that such a measure does not say anything about the distribution throughout the corpus of the characters that form it. If a stem has a variety of 10, we may well have a high number of occurrences of one specific successor and only a few of the nine others. Even worse: a foreign word can occur once in the corpus and impact the variety as much as any other highly frequent word.

Unlike Harris, Andreev has a statistical approach to the morpheme boundary identification problem. As we saw in Section 2.2., the gradient rate is very similar to entropy. Some works (Hafer and Weiss, 1974; Juola et al., 1994) that adopted Harris's principles proposed to replace the simple diversity count by the entropy of the probability distribution over the neighbouring characters. This measure is borrowed from Information Theory and is used to assess the degree of predictability of an event. Here is how Hafer and Weiss (1974) use entropy: $c(\alpha_i)$ is the number of words in the corpus containing the stem identified up to position i in a word α ; $c(\alpha_{ij})$ is the size of the subset of $c(\alpha_i)$ in which the character at position $i+1$ is the j^{th} letter of the alphabet. Thus the probability of the successor of $c(\alpha_i)$ is estimated by $\frac{c(\alpha_{ij})}{c(\alpha_i)}$. The entropy is then computed in the following way (n is the number of letters in the alphabet):

$$H(\alpha_i) = -\sum_{j=1}^{n} \frac{c(\alpha_{ij})}{c(\alpha_i)} \log_2 \frac{c(\alpha_{ij})}{c(\alpha_i)} \tag{5}$$

According to this equation, a high entropy says that all the neighbouring characters have a similar probability, which makes it harder to predict the right one. This situation indicates the presence of a morpheme boundary. In the same way, Andreev's gradient rate is a measure of the predictability of the most frequent neighbouring characters: the lower[12] it is, the harder it is to predict the neighbouring character. Therefore, Andreev is, to our knowledge, the first to have used a way to measure the dispersion of successor/predecessor distribution by means of an equation that shares some properties with entropy. The difference is that Andreev's gradient rate involves only two characters and starts from the informant that is extended to other characters until the border with the stem is reached. He is nevertheless not cited by Hafer and Weiss (1974), who introduce a very similar idea more than ten years later. Note that the same procedure is used today in Goldsmith's *Linguistica*. Goldsmith (2006) uses entropy to check whether a character on the stem side should be on the affix side, as `i` in `attent`**`i`**`-on`.

4.2. Pointwise mutual information

Andreev's correlative function is another measure that is similar to the Information Theory concept of pointwise mutual information, although the author does not mention this similarity. In fact, he gives two distinct definitions of this measure. The first (Section 2.1.) is a means of identifying characters that are the most likely to be part of an affix (the informants) by comparing

[12] A morpheme boundary is indicated either by a higher entropy or, on the contrary, by a lower gradient rate. We suspect that this is the reason why Baklušin (1965) uses the term *negentropy* (or negative entropy) for this last measure, although neither he, nor Andreev, refer to Information Theory.

the probability of a character conditioned on its position in the word to its marginal probability. This bears a ressemblance to the way Church and Hanks (1990) compare the probability of jointly observing two words x and y with the probability of seeing them independently. Equation (6) is the same as the correlative function, except that Andreev does not use a logarithm (see Equation 1).

$$I(x,y) \equiv log_2 \frac{P(x,y)}{P(x)P(y)} = log_2 \frac{P(x|y)}{P(x)} \tag{6}$$

The second definition of the correlative function in Equation (2) is a means of selecting the most likely affixes. A very similar procedure is applied in *Linguistica*, although Goldsmith (2006) views the problem of segmentation from a different level: he includes the search of the paradigm in a global optimization problem. Just as in Andreev's algorithm, a first step consists of heuristics aimed at providing preliminary hypotheses about the way the corpus should be segmented. These hypotheses are then accepted, provided that they minimise the *Description Length*, which is a measure of how concise the representation of the data is given the model, turning the segmentation task into a data compression problem.[13]

The heuristic search for the right segmentation described by Goldsmith (2001) goes from the search for peaks in the words (see 4.1.), up to a distribution over all possible segmentations.[14] He finally proposes to use *weighted mutual information* in order to reach the optimal segmentation. This heuristic approach is described for suffix identification. In Equation (7), borrowed from (Goldsmith, 2001), k-grams of characters are considered, n_k being the word boundary.

$$\text{WMI} = \frac{[n_1,n_2,...,n_k]}{\text{Total k-gram count}} \log \frac{[n_1,n_2,...,n_k]}{[n_1][n_2]...[n_k]} \tag{7}$$

$$\text{CF} = \frac{p(n_1,n_2,...,n_k)}{p(n_1)p(n_2)...p(n_k)} \tag{8}$$

For the sake of comparison, we slightly reformulate Andreev's *correlative function* in Equation (8). Both equations measure how the frequency of an affix differs from a random distribution of the characters composing it, where each character is seen as independent from its position within the word and from the other characters. Finally, both use this idea to extract from the corpus candidate affixes that are the most likely to enter a paradigm. Although these equations slightly differ from the classical definition of mutual information in (Fano, 1961), they still introduce a relation between two perspectives on a set of events; they are, on the one hand, seen as mutually dependent and, on the other hand, as independent.

5. Conclusion

We have described Andreev's statistico-combinatorial model for the unsupervised segmentation of words into morphemes and presented results over three European languages. The main weak points of this approach are the multitude of manually set parameters at various steps (threshold for informant identification, gradient rate threshold, maximum affix length, reduction threshold, reduction coefficient), which are very dependent on the language being analysed and on the corpus size. Despite these points, some of the solutions Andreev proposed, such as entropy measure for morpheme boundary identification and mutual information to select the most relevant affixes, are now used in state-of-the-art systems.

[13] See also (Creutz and Lagus, 2002).
[14] Learned using the Expectation-Maximisation (EM) procedure.

His work suffered from his isolation from the world scientific community; while he never cites Harris or Fano and barely mentions Shannon's work on Information Theory (only once in a footnote), he himself has almost never been cited by posterity. His findings remain nevertheless fundamental in the history of the task.

6. Acknowledgements

This work has been partly funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 645452 (QT21).

References

Andreev, N. D. (1965). *Statistiko-kombinatornoe modelirovanie jazykov*. Nauka.

Andreev, N. D. (1967). *Statistiko-kombinatornye metody v teoretičeskom i prikladnom jazykovedenii*. Nauka.

Andreeva, L. (1963). Statistiko-kombinatornoe vydelenie paradigmy pervogo mofologičeskogo tipa v russkom jazyke. In *Materialy po matematičeskoj lingvistike i mašinnomu pervodu: Sbornik II*, pages 45–60. Izdatel'stvo Leningradskogo universiteta.

Andreeva, L. (1965). Algoritmičeskoe formirovanie razrjada suščestvitel'nogo v russkoj morfologii. In *Statistiko-kombinatornoe modelirovanie jazykov*, pages 27–31. Nauka.

Baklušin, A. (1965). Statistiko-kombinatornoe vydelenie pervogo morpfofonematičeskogo tipa v russkom jazyke. In *Statistiko-kombinatornoe modelirovanie jazykov*, pages 39–48. Nauka.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Creutz, M. and Lagus, K. (2002). Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning - Volume 6*, pages 21–30.

Cromm, O. (1997). Affixerkennung in deutschen wortformen. *LDV Forum*, 14(2):4–13.

Fano, R. M. (1961). Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29:793–794.

Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Goldsmith, J. (2006). An algorithm for the unsupervised learning of morphology. *Natural Language Engineering*, 12(04):353–371.

Goldsmith, J. (2010). Segmentation and morphology. In *The Handbook of Computational Linguistics and Natural Language Processing*, chapter 14. Wiley-Blackwell.

Hafer, M. A. and Weiss, S. F. (1974). Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10:371–385.

Hammarström, H. and Borin, L. (2011). Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Harris, Z. (1955). From phoneme to morpheme. In *Language*, pages 190–222.

Harris, Z. (1967). Morpheme boundaries within words: report on a computer test. In *Transformations and Discourse Analysis Papers*, chapter 14.

Juola, P., Hall, C., Smit, P., and Boggs, A. (1994). Corpus-based morphological segmentation by entropy changes. In *Conference on the Cognitive Science of Natural Language Processing*, Dublin, Ireland.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT.