6

N-Grams Model For Polish

Bartosz Ziółko, Dawid Skurzok
Department of Electronics
AGH University of Science and Technology
Kraków, Poland

1. Introduction

N-grams are very popular in automatic speech recognition (ASR) systems (Young et al., 2005), (Lamere et al., 2004), (Whittaker & Woodland, 2003), (Hirsimaki et al., 2009). They have been found as the most effective models for several languages. N-grams calculated by us will be used for the language model of a large vocabulary Polish ASR system and other outside application, first of them being SnapKeys virtual keyboard. Our earlier results and process of collecting statistics were described already (Ziółko, Skurzok & Ziółko, 2010). In this chapter we want to describe a complete model and its applications.
Creating a large vocbulary model of Polish is a difficult task because there are fewer Polish text corpora then for English. What is more, Polish is very inflected in contrast to English. The rich morphology causes difficulties in training language models due to data sparsity. Much more text data must be used for inflected languages than for positional ones to achieve the model of the same efficiency (Whittaker & Woodland, 2003).

2. Available text corpora for Polish

There are 280 000 words in Polish *myspell* dictionary. The number contains only basic forms. With all inflections, over 1 000 000 words can be easily expected. This is just without proper names. In our case we noted several million words, because of proper names and errors.
The IPI PAN Corpus (Przepiórkowski, 2004) is a the main professional and official corpus of Polish texts. Currently, there are over 250 million segments which are morphosyntactically annotated in a publicly available version. It was developed by the Linguistic Engineering Group at the Institute of Computer Science, Polish Academy of Sciences. The same group works on creating much larger corpus of Polish toegether with some publishers.
However, there are several larger corpora of Polish. They are often not annotated and not available publicly. It is a result of a specific approach of Polish law to copyrights. It is legal to download any texts from Internet, even, if they were put there without authors permission. However, it is not legal to upload any such materials anywhere without persmission and this law is very strictly enforced.
This is why natural language researchers working on Polish do not offer their resources both for free or commercialy, eventhough, some of them collected relatively large data sets. For the mentioned reason, it is not easy to estimate real sizes of corpora of Polish texts.
Newspaper articles in Polish were used as our first corpus. They are Rzeczpospolita newspaper articles taken from years 1993-2002. Several millions of Wikipedia articles in Polish

Corpus	MBytes	Mwords	Perplexity
Rzeczpospolita journal	879	104	8 918
Wikipedia	754	97	16 436
Literature	490	68	9 031
Transcripts	325	32	4 374
Literature 2	6500	949	6181
Literature 3	285	181	4258

Table 1. Analysed text corpora with their sizes, perplexity. More data (websites and literature) were already collected but not analysed yet

Corpus	Basic forms	1-grams	2-grams	3-grams
Rzeczpospolita journal	832 732	856 349	18 115 373	43 414 592
Wikipedia	2 084 524	2 623 358	31 139 080	61 865 543
Literature	610 174	1 151 043	23 830 490	50 794 854
Transcripts	183 363	381 166	6 848 729	16 283 781
Literature 2	?	6 162 530	153 152 158	441 284 743
Literature 3	?	1 229 331	36 297 382	93 751 340

Table 2. The number of different *n*-grams

Corpus	single 1-grams	%	1-grams with errors	%
Rzeczpospolita journal	363 391	42.4	7435	0.86
Wikipedia	379 147	46.5	108 338	4
Literature	467 376	41	75 204	6.5
Transcripts	147 440	39	1 373	0.4
Literature 2	3 552 379	57.6	343 211	5.27
Literature 3	485 713	39.5	6040	0.48

Table 3. Errors in the analysed corpora

made another corpus. The smallest articles were removed from the corpus. In this way we avoided some Wikipedia patterns like *Zawada - a village in Poland, located in Łódzki voivodeship, in Tomaszewski powiat, in Tomaszów Mazowiecki parish. During years 1975-1998, the village belonged to piotrkowskie voivodeship.* There are over 50 000 villages described using exactly same pattern. As a result before we removed them, this pattern provided the list of 5 most common 3-grams, even after combining Wikipedia with two other corpora. Several thousands literature books in Polish from different centuries were used. The fourth corpus is a collection of transcripts from the Polish Parliament, its special investigation committees and Solidarność meetings. They contain mainly transcribed speech but also some comments on the situation in rooms. It is not as big as others but the only one containing transcriptions of spoken language. What is more its topics are law oriented, which corresponds very well with our project, which provides ASR system for Police, administration and other governmental institutions. We are still in the process of collecting more Polish text corpora and combining the statistics of the described ones.

In all cases perplexity is very high comparing to typical English corpora. It is because of inflected nature of Polish and significant number of proper names in the corpora.

3. Problems with processing Polish corpora

Some English, Russian, Chinese and other foreign words appeared in the statistics as well as single letters. Such words could be effects of including some foreign quotes in articles. However, most of the foreign words are proper names and they appeared in Polish sentences. After an analysis of results of collecting n-gram statistics from various corpora, we decided that some supervised correction is necessary. Because of the amount of data, the choice of strategy in this process was crucial from financial point of view. We designed and implemented software Fixgram (Ziółko, Skurzok & Michalska, 2010) to optimise n-gram corrections by time efficiency.

The list of words for corrections is preprepared on a server. This is why, it is partly unsupervised method. Three schemes of preparing words were implemented. The first one is finding pairs of words which are different only by orthographic notation, in example *rz* and *ż*. The second is by finding words with any non-Polish letters. The third method is by comparison with *myspell* dictionary. The words which do not exist in *myspell* are also more likely to be errors then others. A user of Fixgram receives a database of words chosen for corrections to save time spent on automatic search for them in a database during human work. All chosen words are given to the Fixgram user in order by the number of times they appeared in a corpus. All, less common cases will be done automatically, typically by deleting. There is no reason in spending human time for rare cases which are likely to be incorrect and not crucial for statistics. The results from one corpus can be transferred to another one. Sometimes human decisions can be generalised and used for less often cases.

A few types of problems were encountered. The first one are Chinese and English proper names. They appeared quite frequently in the newspaper corpus. Often two Chinese names were detected as orthographic errors because of differences only in *ch* and *h*. Chinese proper names tend to be also often in addition to a Polish word, so one orthographic transcription is for a correct Polish word and the other for Chinese proper name.

Another type of a problem are words which were split into two words with a space so they appeared as two separate words in n-grams. These are difficult to be found automatically.

There are also words which are wrongly formatted (not in UTF-8). Most of them are not in any of known to us standards for Polish letters. This is because we changed all typical standards to UTF-8 before collecting the statistics. These words can still be recognised by a human, as typically there is only one special Polish letter and other are standard Latin letters.

Fixgram (Fig. 1) (Ziółko, Skurzok & Michalska, 2010) presents contexts of each word (2- and 3-grams). It makes correcting these cases much easier. Apart from that, quite a lot of Russian words and single letters (in Cyrillic) were discovered. All of them were removed.

Several automatically detected words were actually correct. For example, there are plenty of similar surnames with an only difference in Polish special ortographic notation. There were some other words which are correct with both orthographic transcriptions but different senses, like morze (Eng. sea) and może (Eng. maybe). These cases were kept in the n-gram database by a human decision.

We have an extra collection of texts from Internet. However, to ensure proper quality, these websites will be first filtered using statistics collected from literature and journals. Only websites with very little new 1-grams will be accepted and added to the model. This process will be repeated iteratively several times. The decisions can be also taken using phoneme statistics of Polish which we also already calculated and currently are improving. In these ways we want to use Internet resources to analyse as much text as possible, but to avoid including texts of low quality or non-Polish ones.

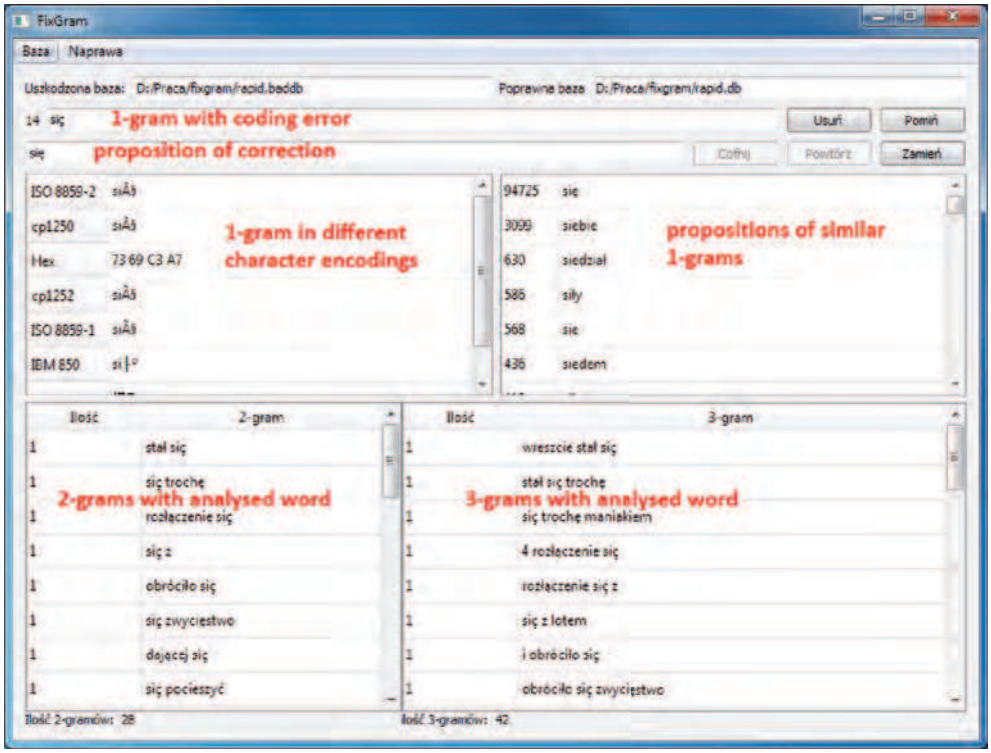


Fig. 1. Screenshot of our Fixgram (Ziółko, Skurzok & Michalska, 2010) software to correct n-gram statistics

4. Results

The most common words in Polish are presented in Table 4. Most frequent 2-grams and 3-grams show Tables 5 and 6. Collected statistics show that the amount of text we used was enough to create representative statistics for 1-grams, 2-grams and even for 3-grams. It is the first such model for Polish.

The most popular 1-grams in Polish are mainly pronouns, what is not surprising. The most popular 2- and 3- grams contain often a dot. Its commonality in the statistics is overhelming but the probability that a particular word starts or ends a sentence is indeed much higher then that two exact words appear next to each other.

The English translations were provided in Table 4 with 1-grams. However, it is quite difficult to translate pronouns without a context. This is why, there are sometimes several translations and eventhough, they are only brief and not complex translations. One of the commonly used words is się. It is a reflexive pronoun. It could be translated as oneself, but it is much more common in Polish than in English. It is used always, if a subject activity is conducted on herself or himself.

The distribution of 1-grams is presented in Fig. 2. The histogram has an expected shape, similarly to histograms of 2- and 3-grams.

word (Eng.)	%	word (Eng.)	%	word (Eng.)	%
.	8.235	kiedy (when)	0.160	niego (him)	0.085
i (and)	2.365	gdy (while)	0.157	jako (as)	0.085
w (in)	0.234	by (would)	0.150	lecz (but)	0.083
się (r.p.)	2.255	ten (this)	0.141	gdzie (where)	0.082
nie (no,not)	1.714	ma (has)	0.139	je (them f.,	0.081
na (on, at)	1.635	który (which m.)	0.138	eats)	
z (with)	1.498	jednak (however)	0.132	nich (them)	0.080
do (to,till)	1.093	ją (her)	0.131	nas (us)	0.078
to (it, this)	0.928	pod (under)	0.129	siebie (themselves)	0.078
że (that)	0.890	była (was f.)	0.129	lub (or)	0.078
a (and)	0.690	przed (before,	0.128	aby (so as)	0.077
o (about, at)	0.549	in front of)	-	te (these f.)	0.076
jak (how,like)	0.485	nawet (even)	0.128	tych (these m.)	0.075
jest (is)	0.440	pan (master)	0.126	pani (madam)	0.075
po (after)	0.426	teraz (now)	0.124	niż (than)	0.074
ale (but)	0.396	ja (I)	0.123	ani (neither)	0.074
co (what)	0.393	bardzo (very)	0.122	(f. prop. name)	-
tak (yes)	0.366	przy (next to)	0.121	można (may)	0.071
za (for,	0.343	są (are)	0.119	nigdy (never)	0.069
behind, by)	-	które (which f. pl.)	0.119	właśnie (just)	0.069
od (from, since)	0.319	tu (here)	0.114	sam (alone)	0.068
jego (his)	0.282	być (be)	0.111	były (were f.)	0.067
przez (through)	0.271	więc (so)	0.110	która (which f.)	0.066
jej (her)	0.262	też (also)	0.107	dobrze (well)	0.065
tym (this)	0.258	tej (this f.)	0.106	niej (her)	0.065
go (him)	0.257	on (he)	0.102	także (also)	0.064
już (yet,	0.252	wszystko (all)	0.101	zawsze (always)	0.063
already)	-	tam (there)	0.101	ty (you)	0.061
tylko (only)	0.230	jeśli (if)	0.101	ta (this f.)	0.060
czy (if)	0.223	nim (him)	0.101	domu	0.060
tego (that, hereof)	0.216	coś (something)	0.101	(house gen.)	-
mnie (me)	0.211	będzie (will be)	0.100	albo (or)	0.060
był (was m.)	0.203	bo (because)	0.099	sposób (way,	0.060
było (was n.)	0.200	nic (nothing)	0.098	method)	-
ze (of, by,	0.190	bez (without)	0.097	oczy (eyes)	0.060
about, with)	-	miał (had)	0.095	jakby (as if)	0.059
mu (him)	0.186	nad (over)	0.094	im (them)	0.059
dla (for)	0.185	żeby	0.094	mam (I have)	0.059
mi (me)	0.182	(in order to)	-	jestem (I am)	0.059
może (maybe)	0.180	ci (you)	0.092	oraz (and)	0.059
sobie (ourself)	0.179	powiedział (said)	0.091	ludzi (people)	0.058
ich (their)	0.178	potem (afterwards)	0.089	raz (one)	0.058
jeszcze (still)	0.169	u (at)	0.086	lat (years)	0.058

Table 4. Top of the 1-gram statistics of Polish collected from literature corpus of 949 371 453 words, (r.p. – reflexive pronoun, m. – masculin, f. – feminine, n. – neuter, pl. – plural, gen. – genitive). Approximated English translations are given in brackets

word (Eng.)	‰	word (Eng.)	%	word (Eng.)	%
chwili (moment)	0.575	głową (head)	0.423	proszę (please)	0.336
aż (till)	0.572	tę (this)	0.423	byli (were)	0.333
ona (she)	0.558	chwilę (moment)	0.411	czego (what)	0.331
wtedy (then)	0.548	dalej (farer)	0.411	pracy (woek)	0.330
no	0.547	ku (towards)	0.405	taki (such)	0.330
więcej (more)	0.543	mój (my)	0.402	ziemi (ground, earth)	0.329
mógł (could)	0.5437	zaś	0.390	czasie (time)	0.327
cię (you)	0.541	innych (others)	0.389	pierwszy (first)	0.327
między (between)	0.540	człowiek (human)	0.387	zaczął (started)	0.327
bardziej (more)	0.539	nikt (noone)	0.386	przykład (example)	0.326
nią (her)	0.531	dlatego (therefore)	0.385	wszystkim (all)	0.325
gdyby (if)	0.528	ktoś (someone)	0.384	człowieka (human)	0.325
roku (year)	0.527	powiedziała (said)	0.383	głos (voice)	0.325
których (which)	0.526	swoje (one's)	0.380	mogę (can, may)	0.324
również (also)	0.520	takie (such)	0.379	jakie (what)	0.324
czasu (time)	0.514	iż	0.378	musi (must)	0.323
wszystkie (all)	0.512	słowa (words)	0.378	temu (this)	0.323
jeden (one)	0.508	później (later)	0.377	prawie (almost)	0.323
wiem (know)	0.500	trochę (little)	0.375	trzy (three)	0.322
czym (what)	0.499	pana (master's)	0.368	znów (again)	0.321
wiele (many)	0.493	tyle (this much)	0.361	chciał (wanted)	0.319
którzy (which)	0.488	życie (life)	0.361	miejsce (place)	0.317
przecież (after all)	0.485	twarz (face)	0.360	myśli (think)	0.316
we (in)	0.481	szybko (fastly)	0.359	panie (sir)	0.314
czas (time)	0.481	końcu (end)	0.355	strony (pages,sides)	0.313
kto (who)	0.480	ponieważ (because)	0.351	obok (next to)	0.313
nam (us)	0.480	naprawdę (really)	0.351	zupełnie (absolutely)	0.313
wszyscy (all)	0.476	cały (whole)	0.350	powiedzieć (to say)	0.313
miała (had)	0.476	niech (let)	0.348	głowę (head)	0.313
kilka (a few)	0.475	jesteś (you are)	0.347	rzekł (said)	0.310
drzwi (doors)	0.473	dopiero (but,until)	0.347	mimo(despite)	0.310
wszystkich(all)	0.471	dzieci (childs)	0.344	nimi (them)	0.308
chyba (actually)	0.462	poza (apart from)	0.344	swego (own)	0.307
razem (together)	0.460	wreszcie (at last)	0.344	wielu(many)	0.306
którym (which)	0.453	którą (which)	0.342	ręce (hand)	0.305
dlaczego (why)	0.453	tutaj (here)	0.342	stronę (page, side)	0.303
której (which)	0.442	zbyt (too)	0.342	wciąż (still)	0.301
ludzie (people)	0.438	znowu (again)	0.341	coraz	0.301
nagle (suddenly)	0.438	oczywiście (of course)	0.341	moje (my)	0.297
dwa (two)	0.438	jeżeli (if)	0.341	dzień (day)	0.295
którego (which)	0.432	rzeczy (things)	0.340	pokoju (room, peace)	0.294
trzeba (need)	0.430	dnia (day)	0.340	mają (have)	0.294
choć (however)	0.429	jakiś (some)	0.337	każdy (each)	0.291
życia (life)	0.428	podczas (during)	0.337	prawda (true)	0.290
sobą (self)	0.428	ciebie (you)	0.336	został (became)	0.288

2-gram	‰	2-gram	‰	2-gram	‰
. nie	3.13	. jest	0.32	. dlaczego	0.20
. w	2.40	a potem	0.32	w którym	0.20
. a	1.69	. do	0.31	że jest	0.20
się w	1.51	mu się	0.31	. tylko	0.20
. to	1.49	w tej	0.31	. zapytał	0.20
. ale	1.24	. teraz	0.31	na pewno	0.20
. i	1.19	to co	0.30	na niego	0.20
się na	1.12	w końcu	0.29	i na	0.20
się z	1.05	do tego	0.29	po czym	0.20
. na	1.03	na przykład	0.29	jak to	0.20
się do	1.02	z tego	0.29	do domu	0.19
. tak	0.80	tak .	0.29	a nie	0.19
. z	0.74	z nich	0.29	. nic	0.19
. czy	0.71	po prostu	0.28	w ogóle	0.19
. po	0.71	. był	0.27	z tym	0.19
. co	0.68	to jest	0.27	co .	0.19
w tym	0.66	. za	0.27	nie mógł	0.19
się .	0.66	co się	0.26	nie był	0.19
się że	0.61	że w	0.26	. nawet	0.19
że nie	0.57	to że	0.26	się za	0.19
. jak	0.54	i tak	0.26	w ten	0.19
nie ma	0.53	i z	0.25	po raz	0.18
o tym	0.52	się o	0.25	nie może	0.18
. kiedy	0.50	się od	0.25	jak się	0.18
i nie	0.47	. potem	0.24	siebie .	0.18
się nie	0.46	. od	0.24	jeden z	0.18
się i	0.44	nic nie	0.24	. mam	0.18
nie jest	0.41	jest to	0.24	domu .	0.17
. o	0.41	nie tylko	0.24	. niech	0.17
to nie	0.41	. jego	0.24	jest w	0.17
. może	0.41	nie wiem	0.23	się to	0.17
na to	0.40	tak jak	0.23	. on	0.17
i w	0.40	. przez	0.23	w czasie	0.17
. no	0.40	w jego	0.23	do mnie	0.17
nie było	0.39	mnie .	0.22	. jestem	0.17
. jeśli	0.39	się po	0.22	nie będzie	0.17
ale nie	0.37	z nim	0.22	. oczywiście	0.17
nie .	0.37	i to	0.22	. w stronę	0.17
mi się	0.37	a w	0.21	a więc	0.17
że to	0.35	do niego	0.21	jak i	0.17
nigdy nie	0.34	głową .	0.21	po chwili	0.17
. gdy	0.34	. ten	0.21	. była	0.17
. ja	0.33	. już	0.21	w nim	0.17

Table 5. Top of the 2-gram statistics of Polish from a literature corpus

2-gram	‰	2-gram	‰	2-gram	‰
nikt nie	0.166	tym razem	0.142	z pewnością	0.124
. proszę	0.163	się tak	0.142	z nią	0.124
. spytał	0.162	na nią	0.142	wszystko co	0.123
. wszystko	0.162	od razu	0.142	. wiem	0.123
już nie	0.161	. więc	0.141	tym samym	0.122
. bo	0.161	i jego	0.141	z jego	0.120
na tym	0.160	tego co	0.141	powiedział .	0.120
ten sposób	0.160	poza tym	0.141	wcale nie	0.119
na mnie	0.160	ze sobą	0.140	. nagle	0.119
co to	0.159	go w	0.139	drzwi .	0.119
. jeżeli	0.158	to było	0.137	dlatego że	0.118
to .	0.157	. wszyscy	0.137	. dlatego	0.118
a ja	0.156	przez chwilę	0.137	a teraz	0.118
. nigdy	0.156	. jej	0.137	. miał	0.117
do siebie	0.155	aż do	0.137	się przez	0.117
nie miał	0.155	z powrotem	0.136	jak na	0.117
się jak	0.155	było to	0.136	co do	0.117
w stanie	0.154	dla mnie	0.135	w każdym	0.117
do niej	0.154	w ciągu	0.134	sobie że	0.117
na jego	0.153	. lecz	0.134	ale w	0.116
spojrzał na	0.153	nie można	0.134	. ty	0.116
za to	0.153	się nad	0.134	. nikt	0.116
. jeszcze	0.153	ale to	0.133	tak samo	0.115
wraz z	0.151	a może	0.133	ludzi .	0.115
może być	0.150	. pan	0.133	za nim	0.115
o to	0.150	ze mną	0.133	się stało	0.115
. gdyby	0.150	to w	0.133	nie była	0.115
czy nie	0.148	. jednak	0.132	niego .	0.114
to wszystko	0.148	w jej	0.132	jak w	0.114
. chyba	0.146	i że	0.132	. wtedy	0.114
czy to	0.146	. było	0.131	lat .	0.113
uśmiechnął się	0.146	a co	0.130	w kierunku	0.113
się ze	0.146	nie mam	0.129	w niej	0.112
przede wszystkim	0.145	. dobrze	0.129	podobnie jak	0.112
tym że	0.145	. kto	0.128	w sobie	0.112
jest .	0.145	. dla	0.127	odezwał się	0.112
nie mogę	0.145	jeszcze nie	0.127	już w	0.111
w domu	0.144	dobrze .	0.126	go do	0.111
. przecież	0.144	. ze	0.126	z tych	0.111
być może	0.144	. ta	0.125	w której	0.111
oczy .	0.143	. bardzo	0.125	życia .	0.111
prawda .	0.143	się już	0.125	nadzieję że	0.110
tego nie	0.143	a także	0.124	dalej .	0.110
był to	0.142	to znaczy	0.124	. gdzie	0.110

2-gram	‰	2-gram	‰	2-gram	‰
. mimo	0.109	się pod	0.970	wszystko .	0.889
tej chwili	0.109	w takim	0.969	. przed	0.888
. przy	0.109	że się	0.969	. zawsze	0.887
nawet nie	0.109	do tej	0.966	. mój	0.886
z powodu	0.109	w porządku	0.966	to samo	0.884
a to	0.108	. jesteś	0.962	nim .	0.880
go .	0.108	. tym	0.961	. powiedział	0.879
. cóż	0.107	. coś	0.956	ale i	0.878
to się	0.107	o czym	0.952	. te	0.877
. tu	0.106	o czym	0.952	od czasu	0.875
można było	0.106	na chwilę	0.950	ziemi .	0.872
w życiu	0.106	. sam	0.950	jak gdyby	0.870
z nimi	0.106	. tam	0.945	ci się	0.870
raz pierwszy	0.105	chodzi o	0.939	. dopiero	0.867
i co	0.105	to był	0.939	podczas gdy	0.867
a na	0.104	. są	0.939	. muszę	0.865
odwrócił się	0.104	. pod	0.938	. jeden	0.864
tym co	0.103	. poza	0.935	było .	0.864
. trzeba	0.103	nie są	0.933	w pobliżu	0.862
i po	0.103	razem z	0.932	. bez	0.861
w dół	0.103	na ziemi	0.932	i do	0.860
wiem .	0.103	o co	0.929	życie .	0.858
się jej	0.102	zgodnie z	0.929	zrobić .	0.858
od tego	0.102	z tobą	0.928	jeszcze raz	0.856
na temat	0.102	się jeszcze	0.927	na siebie	0.853
a nawet	0.101	za sobą	0.923	więcej niż	0.852
ja .	0.101	był w	0.920	w których	0.847
po co	0.101	to na	0.919	. spytała	0.843
do nich	0.101	do końca	0.918	. wreszcie	0.841
w górę	0.101	sobie sprawę	0.918	tylko w	0.841
. właśnie	0.100	to tylko	0.917	co z	0.841
względu na	0.100	myślę że	0.914	to z	0.838
się przed	0.100	by się	0.913	stało się	0.838
była to	0.100	. ona	0.908	. tego	0.836
go na	0.100	nie mogła	0.908	się tylko	0.834
wiedział że	0.100	i o	0.905	że na	0.833
jednym z	0.099	. pani	0.903	. albo	0.830
przy tym	0.099	jednak nie	0.901	po to	0.829
go nie	0.099	tak się	0.900	i jak	0.827
. och	0.098	dla niego	0.900	między innymi	0.823
na jej	0.098	tak że	0.893	mimo to	0.823
nie jestem	0.098	czy też	0.893	i ja	0.822
jeśli nie	0.097	stało .	0.891	w polsce	0.819
to już	0.097	. ponieważ	0.890	w swoim	0.815

3-gram	‰	3-gram	‰	3-gram	‰
w ten sposób	1.71	się z nim	0.613	. dlaczego .	0.480
. tak .	1.59	. no i	0.608	do siebie .	0.480
. nie .	1.55	. nie mogę	0.607	w tym momencie	0.479
. nie wiem	1.32	. nie mam	0.598	. nic nie	0.477
. w tym	1.30	od czasu do	0.593	to nie jest	0.475
. nie ma	1.23	czasu do czasu	0.592	że nie ma	0.474
po raz pierwszy	1.13	w tym samym	0.592	po drugiej stronie	0.473
. to nie	1.10	. tym razem	0.589	. to co	0.472
. w końcu	1.07	o tym że	0.585	w tym czasie	0.472
. ale nie	1.05	sobie sprawę że	0.582	w ogóle nie	0.470
. a więc	0.998	. w każdym	0.573	. a jeśli	0.469
w tej chwili	0.985	. na pewno	0.572	. w takim	0.468
. a co	0.945	. i to	0.572	. przez chwilę	0.468
. czy to	0.911	. a ja	0.565	. po co	0.466
na to że	0.901	. i nie	0.559	. co .	0.464
. a może	0.821	w takim razie	0.552	. i co	0.461
w każdym razie	0.813	. nie nie	0.5525	. nie jestem	0.460
. po chwili	0.811	się do niego	0.550	. a ty	0.457
. poza tym	0.807	w jaki sposób	0.546	nie ma .	0.457
. nigdy nie	0.782	. nikt nie	0.539	do tej pory	0.446
. nie było	0.775	wydaje mi się	0.529	. wiem że	0.446
mi się że	0.764	w porządku .	0.528	. jak się	0.445
. a teraz	0.762	. na przykład	0.524	. a jednak	0.442
. był to	0.757	w stosunku do	0.516	. to był	0.442
do domu .	0.751	mam nadzieję że	0.513	. mam nadzieję	0.441
. być może	0.739	. w ten	0.510	. niech pan	0.437
. w tej	0.733	. tak więc	0.507	o tym .	0.435
. jest to	0.725	. to znaczy	0.507	. mimo to	0.434
ze względu na	0.710	. no to	0.506	. nie chcę	0.431
co się stało	0.707	tak samo jak	0.506	co się dzieje	0.429
. co to	0.704	. a to	0.506	. no cóż	0.428
. a potem	0.703	. była to	0.505	do wniosku że	0.416
. po prostu	0.688	okazało się że	0.504	się z nią	0.408
. myślę że	0.676	. jeden z	0.501	. nie jest	0.403
. ale to	0.660	. było to	0.497	się do niej	0.399
. co się	0.657	w związku z	0.497	o tym nie	0.399
. i tak	0.657	z drugiej strony	0.496	za każdym razem	0.398
się stało .	0.653	zwrócił się do	0.495	. spojrzał na	0.394
nie wiem .	0.641	nie było .	0.489	na pewno nie	0.393
. to jest	0.639	. czy nie	0.488	. uśmiechnął się	0.385
. jak to	0.626	. to było	0.486	. po raz	0.320

Table 6. Top of the 3-gram statistics of Polish from a literature corpus. They are very good data to model language but are difficult to be collected for inflected languages in amount which is enough for applications. The model we manage to build seems to be large enough to properly describe language by statistics of 3-grams

3-gram	‰	3-gram	‰	3-gram	‰
z tego co	0.319	. wszystko to	0.282	się na to	0.252
jeśli chodzi o	0.319	to wszystko .	0.282	i tak dalej	0.251
. wiedział że	0.319	. i w	0.282	. ale co	0.251
. może to	0.317	się w nim	0.281	się o tym	0.250
po to by	0.317	. o co	0.281	się o tym	0.250
na ziemię .	0.317	. cit .	0.280	w każdej chwili	0.250
. tak to	0.316	. ale ja	0.279	. a przecież	0.249
. to wszystko	0.316	to prawda .	0.278	. nie tylko	0.248
odwrócił się i	0.316	jak to się	0.276	. okazało się	0.247
. to prawda	0.315	w gruncie rzeczy	0.276	względu na to	0.247
się ze mną	0.315	. podobnie jak	0.275	się w jego	0.247
. dobrze .	0.314	. ja nie	0.274	udało mi się	0.245
. odwrócił się	0.312	mu się że	0.274	pokręcił głową .	0.244
udało mu się	0.311	w porównaniu z	0.274	. po pierwsze	0.241
że jest to	0.310	. z drugiej	0.271	. no tak	0.241
. oczywiście .	0.309	na ziemi .	0.271	w dalszym ciągu	0.241
za to że	0.308	z tego powodu	0.270	. a zatem	0.241
przed naszą erą	0.308	w chwili gdy	0.269	. nie to	0.241
nie da się	0.308	w dół .	0.269	. spojrzała na	0.241
. nie miał	0.307	się dzieje .	0.268	. od czasu	0.240
. ja .	0.307	na przykład w	0.267	na zewnątrz .	0.240
to znaczy że	0.306	. w jego	0.267	się z tobą	0.239
. jeśli nie	0.304	. zdaje się	0.266	po co .	0.239
. dlaczego nie	0.303	. oczywiście że	0.259	z dala od	0.238
się że to	0.301	. przede wszystkim	0.259	sobie sprawę z	0.238
w tym miejscu	0.301	obawiam się że	0.259	. nawet nie	0.238
do czynienia z	0.301	. uśmiechnęła się	0.258	. przykro mi	0.237
. to była	0.300	. chyba nie	0.258	. to właśnie	0.237
się w stronę	0.299	z nich .	0.258	na to .	0.237
po prostu nie	0.298	że nie jest	0.258	się do nich	0.236
z tego że	0.297	o tym jak	0.258	. nie rozumiem	0.236
na mnie .	0.297	nie można było	0.258	wygląda na to	0.236
nie wiem czy	0.295	z nich nie	0.257	co chodzi .	0.235
co to za	0.294	w taki sposób	0.257	. zgodnie z	0.234
się w tym	0.293	wpatrywał się w	0.256	. naprawdę .	0.234
to jest .	0.293	się nad tym	0.255	jest w stanie	0.233
na niego .	0.292	do drzwi .	0.255	. co ty	0.232
że to nie	0.290	. jeszcze nie	0.255	. i wtedy	0.232
. o ile	0.290	tylko dlatego że	0.254	do niej .	0.231
. nie był	0.289	i spojrzał na	0.254	tej samej chwili	0.231
. sądzę że	0.289	z powrotem .	0.253	. wydaje mi	0.230
za nim .	0.289	w tej sprawie	0.253	i z powrotem	0.230
. nie można	0.285	w przeciwieństwie do	0.253	. do tego	0.229
. a kiedy	0.284	się z nimi	0.253	odwrócił się do	0.228
z jednej strony	0.284	ale to nie	0.252	. nie sądzę	0.228
nie wiem co	0.282	nie mógł się	0.252	to samo .	0.228

3-gram	‰	3-gram	‰	3-gram	‰
z pewnością nie	0.228	a co z	0.210	. wygląda na	0.195
. spytał .	0.228	spojrzał na nią	0.210	w jednym z	0.195
potrząsnął głową .	0.228	. dlatego też	0.209	w odniesieniu do	0.194
nie ma w	0.227	. co prawda	0.209	w jakiś sposób	0.193
się coraz bardziej	0.227	nigdy się nie	0.208	w zależności od	0.192
wydaje się że	0.226	. a czy	0.208	. nie mogła	0.192
po tym jak	0.226	przez jakiś czas	0.207	po raz ostatni	0.192
czy nie .	0.225	. o czym	0.207	związku z tym	0.192
nie było to	0.225	o to że	0.207	. co z	0.191
. tak się	0.224	. wcale nie	0.207	zdawało się że	0.191
z tobą .	0.224	. no .	0.206	. nie wolno	0.191
w górę .	0.224	na myśli .	0.206	. wiem .	0.190
wydawało mi się	0.224	na zawsze .	0.206	. po czym	0.190
za późno .	0.224	. no więc	0.206	do przodu .	0.190
nie jest w	0.224	na to co	0.206	po raz drugi	0.190
. obawiam się	0.222	nie żyje .	0.205	do pracy .	0.190
co to znaczy	0.222	. co do	0.205	. na tym	0.189
nie ma nic	0.222	. to ja	0.204	. o tym	0.189
po obu stronach	0.221	w miarę jak	0.204	tylko po to	0.189
nie było w	0.221	. ale jak	0.203	. ale .	0.189
nie wiem jak	0.220	. tak czy	0.203	to zrobić .	0.188
. wydaje się	0.219	nic z tego	0.202	. to się	0.188
. to już	0.219	się w niej	0.202	że nigdy nie	0.188
był w stanie	0.218	. ale teraz	0.202	do tego że	0.187
. za to	0.218	po to żeby	0.202	się nie stało	0.186
. myślałem że	0.218	. to tylko	0.202	. zdziwił się	0.186
jak na przykład	0.217	. po kilku	0.202	. nie była	0.186
znalazł się w	0.217	. przecież to	0.200	na miejscu .	0.185
. na to	0.217	. to bardzo	0.200	. nie możemy	0.185
coś w rodzaju	0.217	i w tym	0.200	o tej porze	0.185
ze sobą .	0.216	raz po raz	0.199	. przez cały	0.185
na świecie .	0.216	z punktu widzenia	0.199	wyglądało na to	0.185
co się z	0.215	zbliżył się do	0.199	. zastanawiał się	0.185
spojrzała na niego	0.215	znajduje się w	0.199	na drugą stronę	0.184
. gdyby nie	0.215	to co się	0.199	w ostatniej chwili	0.184
. no dobrze	0.214	z powrotem na	0.199	. a poza	0.184
z nim .	0.213	do głowy .	0.199	w milczeniu .	0.184
. wydawało się	0.213	w tym celu	0.199	. tak samo	0.184
z całą pewnością	0.213	. co więcej	0.199	i w ogóle	0.184
i tak nie	0.213	. kiedy się	0.198	. nie mogłem	0.183
wszystko w porządku	0.212	nie mam pojęcia	0.198	. nie będę	0.183
z tego .	0.212	się z tego	0.197	co z tego	0.183
na niego z	0.212	się że w	0.197	co do tego	0.183
nie był w	0.211	. ale czy	0.196	. chodzi o	0.183
na to nie	0.211	w głowie .	0.195	. ale przecież	0.182
. myślisz że	0.210	na wszelki wypadek	0.195	niezależnie od tego	0.182

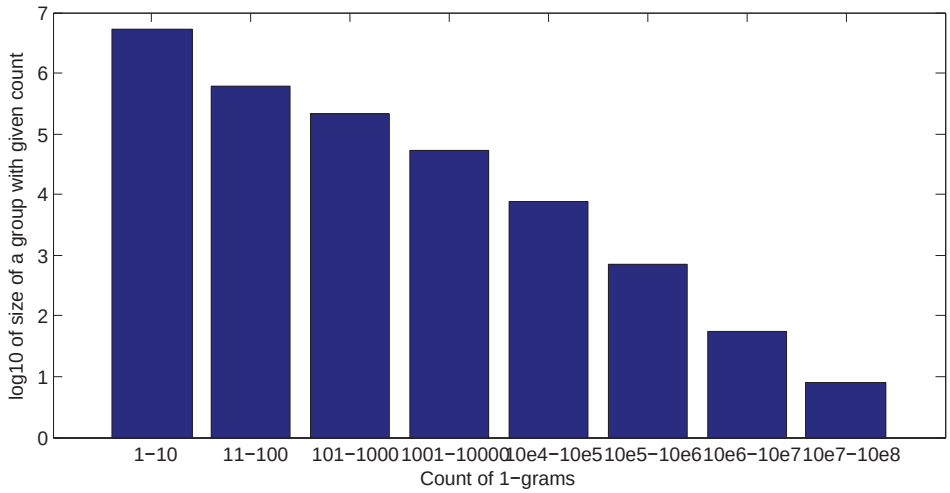


Fig. 2. Histogram of 1-grams (in logarithm scale). There are many 1-grams which are very rare. The amount goes down with increasing count of a 1-gram. The histograms of 2- and 3-grams are very similar

5. Implementation and applications

Storing large vocabulary n-gram model is another issue to concern. 2- and 3-grams cannot be stored as strings because they would use too much disk space. This is why each 1-gram (unigram on Fig. 3) has an ID. The 2-grams are stored as two 1-gram ID, which are integer numbers. The each 2-gram has its id_bigram, so 3-grams are stored as a set of two id_bigrams. The language properties have been very often modelled by n-grams (Huang & Lippman, 1988), (Young et al., 2005), (Manning, 1999), (Jurafsky & Martin, 2008), (Khudanpur & Wu, 1999), (Whittaker & Woodland, 2003), (Hirsimaki et al., 2009). Let us assume the word string $w \in W$ consisting of n words $w_1, w_2, w_3, ..., w_n$. Let $P(W)$ be a set of probability distributions over possible word strings W that reflects how often $w \in W$ occurs. It can be decomposed as

$$P(w) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)...P(w_n|w_1, ..w_{n-1}). \quad (1)$$

It is theoretically justified and practically useful assumption that, $P(w)$ dependence is limited to n words backwards. Probably the most popular are trigram models where $P(w_i|w_{i-2}, w_{i-1})$, as a dependence on the previous two words is the most important, while model complication is not very high. Such models still need statistics collected over a vast amount of text. As a result many dependencies can be averaged. Simplified case of applying n-grams in speech recognition is presented in Fig. 4.

N-grams are the most basic and common language model in ASR systems (Young et al., 2005), (Lamere et al., 2004), (Whittaker & Woodland, 2003), (Hirsimaki et al., 2009). It is a result of their simplicity end effectiveness Our attempt was to build such model for large vocabulary Polish applications. The large number of analysed texts will allow us to predict words being recognised and improve recognition of the ASR system highly.

Polish is highly inflected in comparison to English. The rich morphology causes difficulties in training language models due to data sparsity. Much more text data must be used for inflected

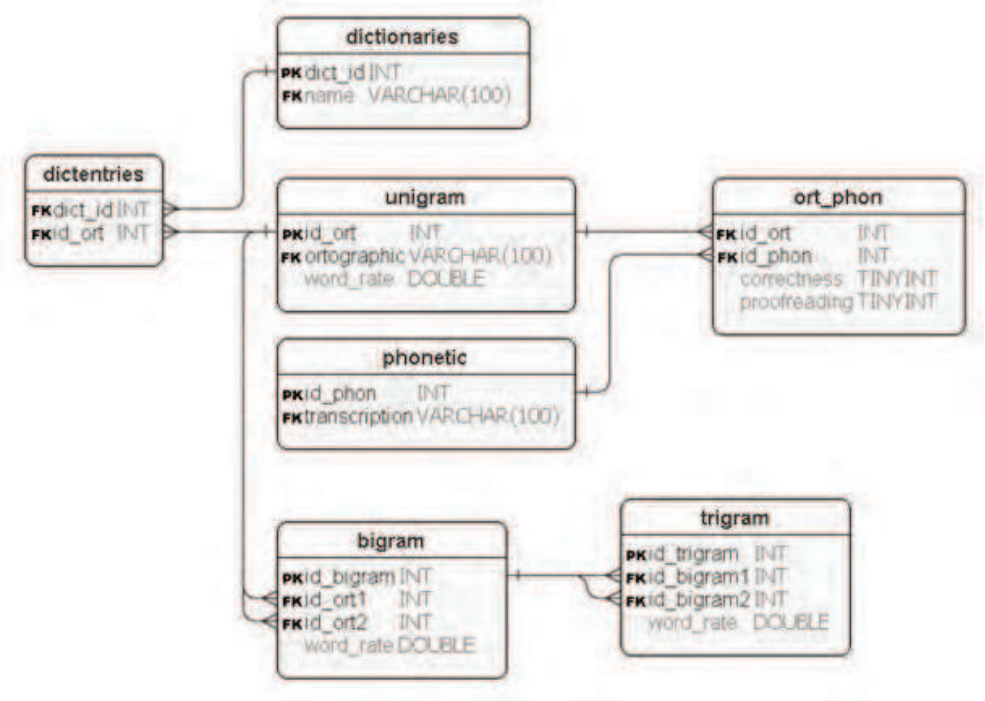


Fig. 3. Our n-gram model is a part of a dictionary implemented in SQL.

languages than for positional ones to achieve the model of the same efficiency (Whittaker & Woodland, 2003).

The modified weighted Levenshtein distance (MWLD) (Ziółko, Gałka, Skurzok & Jadczyk, 2010) and dynamic time warping (DTW) (Rabiner & Juang, 1993) algorithms allow to evaluate a distance of words from an ASR system dictionary with a sequence of phoneme hypotheses. In case of recognising continuous speech, this procedure have to be repeated hundreds thousands of time for different words and different phoneme hypotheses. An optimal decision is taken to find a sequence of word hypotheses. This processes is known as level builder.

Typically, the situation is even more complex. Instead of a sequence of words, a lattice of words should be built. The final sentence hypothesis is taken from the lattice, by applying syntax and semantic modelling.

Word hypotheses are sorted by natural logarithms of MWLD or DTW. The W words with lowest distances are introduced to the lattice for each allowed start point of a word.

Let us assume a set of I word hypotheses and matrix $H \in (C, \Re)^{n \times k}$ of phoneme hypotheses where C stands for a set of characters representing Polish phonemes, $\Re$ are logaritms of propabilities, n is size of C (number of possible phoneme types) and k corresponds to time. Let us introduce w_m as m-th word of a M size $(0 < m \leq M)$ dictionary. Then, let us denote a_i as a start time of ith word hypothesis and b_i as its end. Let us introduce $p_m(a_i = t_1, b_i = t_2)$ as a probability that word w_m is an ith observation for a sequence of phonemes from time a_i to

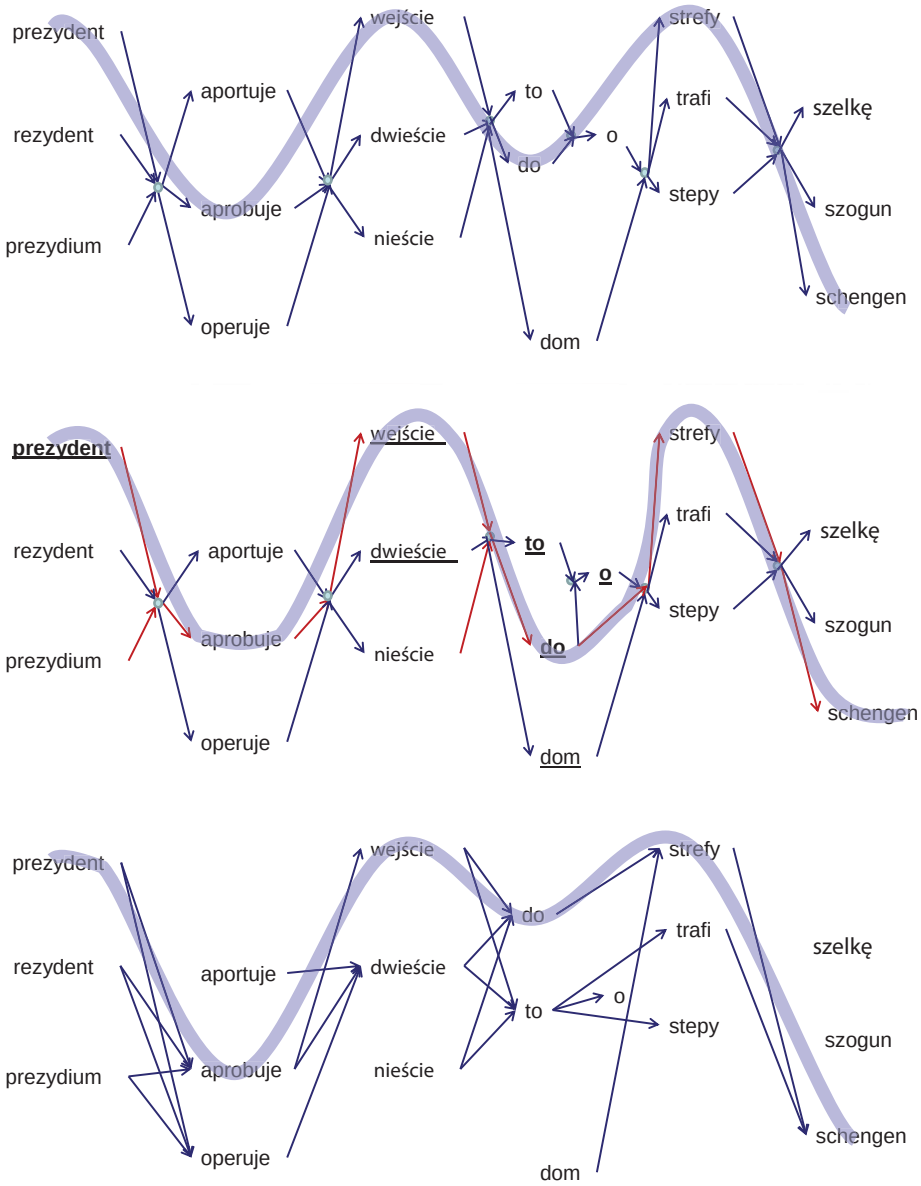


Fig. 4. The general word lattice is presented in the upper diagram. A lattice with stressing of probable 1 grams (bold and undelined) and 2 grams (red arrows) is depicted in the middle one. A word lattice with reduction of unprobable 2-grams is shown in the bottom one. In all cases the correct sentence is marked by a purple shadow. In the second case it leads mainly via strong n-grams. In the third case the proper path still exists in the lattice after reductions

Fig. 5. Real word lattice generated by AGH ASR system shows complexity of the graph and importance of applying language modelling like n-grams

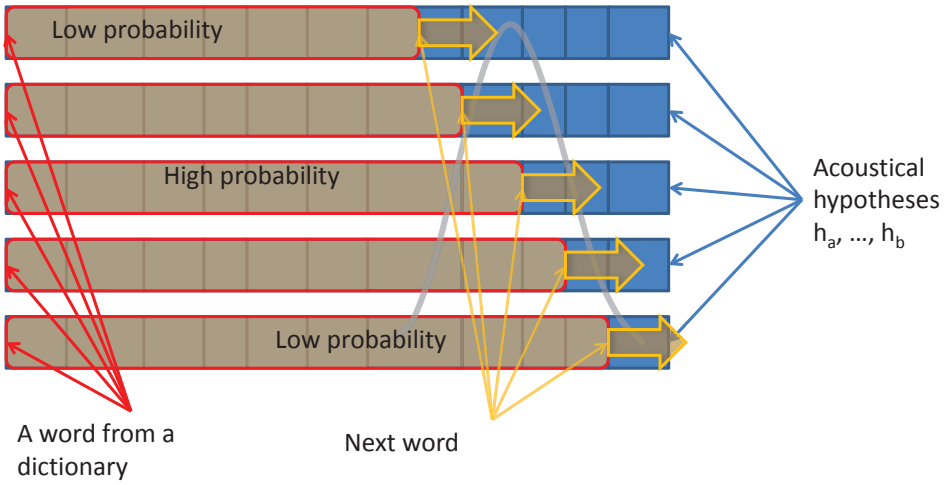


Fig. 6. A level builder fits a dictionary word into acoustical hypotheses on different time scales

time b_i such as

$$a_i = \begin{cases} 1 & \text{for the words following a starting node} \\ b_{i-j} + 1 \pm t & \text{for others} \end{cases}, \tag{2}$$

where b_{i-j} is an end of another word hypothesis and where $t = 3$ is a threshold of allowed time distance between neighbouring words counted in the number of frames (phoneme hypotheses). In the simplest case $j = 1$, but generally $j < i$ (in case of a lattice). The task of level building is to maximise $p_m(a_i = t_1, b_i = t_2)$ by changing m, a_i and b_i. Difference $b_i - a_i$ is constant for a particular word w_m and there are restrictions for a_i described above (a_i of a word has to follow b_{i-j} of another word in time domain). Typically there are between 10

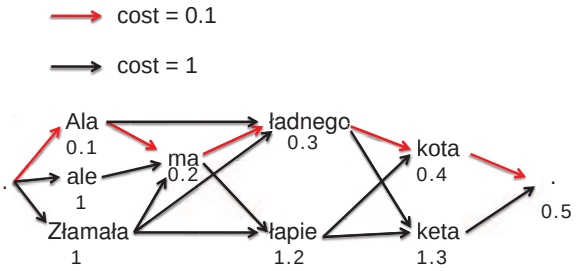
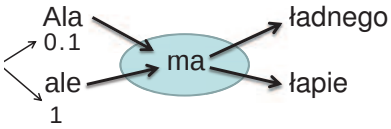


Fig. 7. Simple example of word network showing usage of 2-grams to find the best path. The words in the lattice mean: Ala – female name, ale – but, Złamała – broke (feminine), ma – has, ładnego – pretty (masculin), łapie – catches, kota – a cat (accusativus), keta – a chain (in silesian dialect)



next	cost
ładnego	0.2
łapie	1.1

Fig. 8. Example of calculating weights for a word using 3-grams. Its possible weights are based on words preceding and following in the word network. English translations are in Fig. 7

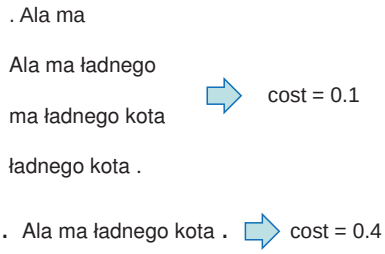


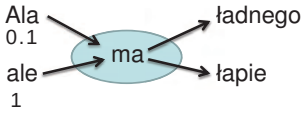
Fig. 9. 3-grams used to decode a sentence from the example from Fig. 8. English translations are in Fig. 7

3-gram	cost
. Ala ma	0.1
. Ala ładnego	1

next	previous	cost
ładnego	.	1
ma	.	0.1

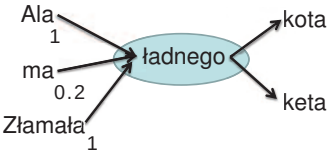
3-gram	cost
. ale ma	1

next	previous	cost
ma	.	1



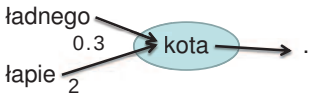
3-gram	cost
Ala ma ładnego	0.1
ale ma ładnego	0.1
Ala ma łapie	1
ale ma łapie	1

next	previous	cost
ładnego	Ala	0.1+0.1 = 0.2
	ale	1 + 0.1 = 1.1
łapie	Ala	0.1 + 1 = 1.1
	ale	1 + 1 = 2



3-gram	cost
Ala ładnego kota	1
Ala ładnego keta	1
ma ładnego kota	0.1
ma ładnego keta	1
Złamała ładnego kota	1
Złamała ładnego keta	1

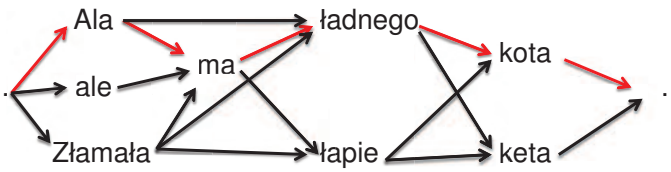
next	previous	cost
kota	Ala	1 + 1 = 2
	ma	0.2 + 0.1 = 0.3
	Złamała	1 + 1 = 2
keta	Ala	1 + 1 = 2
	ma	1 + 1 = 2
	Złamała	1 + 1 = 2



3-gram	cost
ładnego kota .	0.1
łapie kota .	1

next	previous	cost
.	ładnego	0.3 + 0.1 = 0.4
	łapie	2 + 1 = 3

Fig. 10. Proces of finding the best path through the word network using 3-gram weights, node after a node. English translations are in Fig. 7



10 nodes
17 edges
26 possible 3-grams

Fig. 11. Result of searching for the best path through the word network using 3-gram weight, node after a node (see Fig. 10). English translations are in Fig. 7

and 50 parallel word hypotheses allowed to start from a particular time point in the described way.

The word hypotheses are turned into a lattice by connecting nodes if ends and starts are closer to each other in time then a chosen threshold.

Created word lattices are large, which makes searching for a best path time consuming, while ASR system should work in real time. This is why, edges which statistically were found unlikely by n-grams can be cut out.

Finding the best path can be provided using Dijkstra algorithm (Dijkstra, 1959). Applying 2-grams is very straightforward, but using 3-grams is more complex. This is why we will disscuss its possible implementation considering an example. The whole network of our example is presented in Fig. 7, but with simplified values from 2-grams only. Then calculating probability for a particular word using 3-grams is presented in Fig. 8. It has to be stressed that many more calculations have to be conducted to calculate these weights, and also many more values have to be kept when the best path is searched. Fig. 9 shows the entire sentence we want to decode and its weights using 3-grams being components of this sentence. Fig. 10 shows searching the best path node after a node. Our example has 10 nodes and 17 edges. It results in 26 possible 3-grams (Fig. 11).

Typically n-grams of higher orders are smoothed by backing-off methods (Kneser & Ney, 1995; Ney et al., 1994). It can improve results by up to 5%. Another recently popular method is to apply Bloom filter (Bloom, 1970) instead of backing-off.

The presented n-gram model of Polish will be licensed to be available for both research and commercial applications. The first commercial usage will be an Imaginary Interface made by SnapKeys. It has 4 imaginary letter keys at the begining. Afterwards a user can hide them because they can begin to blind type anywhere on the screen. It leaves entire scrren for displaying output data and allow faster typing thanks to smaller finger movements. The interface connects several probability models to find words which a user wants – 1-gram being one of them. The Polish version is now being developed using our model.

6. Conclusions

N-gram models are strightforward but very effective in language modelling. Large corpora are necessary to build effective n-grams models. This and other problems make this task especially complicated for languages like Polish which are highly inflected and without very

large professional text corpora. Eventhough this difficulties, a succesful n-grams model of Polish was build at AGH and offered to public.

7. Acknowledgments

This work was supported by MNISW grant number OR00001905.

8. References

Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors, *Communications of the ACM* **13, 7**.

Dijkstra, E. (1959). A note on two problems in connexion with graphs, *Numerische Mathematik* 1: 269–271.

Hirsimaki, T., Pylkkonen, J. & Kurimo, M. (2009). Importance of high-order n-gram models in morph-based speech recognition, *IEEE Transactions on Audio, Speech and Language Processing* 17(4): 724–32.

Huang, W. & Lippman, R. (1988). Neural net and traditional classifiers, *Neural Information Processing Systems, D. Anderson, ed.* pp. 387–396.

Jurafsky, D. & Martin, J. H. (2008). *Speech and Language Processing, 2nd Edition*, Prentice-Hall, Inc., New Jersey.

Khudanpur, S. & Wu, J. (1999). A maximum entropy language model integrating n-grams and topic dependencies for conversational speech recognition, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Phoenix, AZ* .

Kneser, R. & Ney, H. (1995). Improved backing-off for m-gram language modelling, *Proceedings of International Conference on Acoustics, Speech, and Signal Processing, ICASSP* pp. 181–184.

Lamere, P., Kwok, P., Gouvea, E., Raj, B., Singh, R., Walker, W. & Wolf, P. (2004). The CMU Sphinx-4 speech recognition system, *Sun Microsystems* .

Manning, C. D. (1999). *Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge, MA.

Ney, H., Essen, U. & Kneser, R. (1994). On structuring probabilistic dependencies in stochastic language modelling, *Computer Speech and Language* 8: 1–38.

Przepiórkowski, A. (2004). *The IPI PAN Corpus: Preliminary version*, IPI PAN, Warszawa.

Rabiner, L. & Juang, B. H. (1993). *Fundamentals of speech recognition*, PTR Prentice-Hall, Inc., New Jersey.

Whittaker, E. & Woodland, P. (2003). Language modelling for Russian and English using words and classes, *Computer Speech and Language* 17: 87–104.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. & Woodland, P. (2005). *HTK Book*, Cambridge University Engineering Department, UK.

Ziółko, B., Gałka, J., Skurzok, D. & Jadczyk, T. (2010). Modified weighted Levenshtein distance in automatic speech recognition, *Proceedings of XVI KKZMBM* pp. 116–120.

Ziółko, B., Skurzok, D. & Michalska, M. (2010). Polish n-grams and their correction process, *Proceedings of The 4th International Conference on Multimedia and Ubiquitous Engineering (MUE 2010), Cebu, Philipines* .

Ziółko, B., Skurzok, D. & Ziółko, M. (2010). Word n-grams for polish, *The Tenth IASTED International Conference on Artificial Intelligence and Applications, AIA 2010* .

Speech and Language Technologies

Edited by Prof. Ivo Ipsic

ISBN 978-953-307-322-4
Hard cover, 344 pages
Publisher InTech
Published online 21, June, 2011
Published in print edition June, 2011

This book addresses state-of-the-art systems and achievements in various topics in the research field of speech and language technologies. Book chapters are organized in different sections covering diverse problems, which have to be solved in speech recognition and language understanding systems. In the first section machine translation systems based on large parallel corpora using rule-based and statistical-based translation methods are presented. The third chapter presents work on real time two way speech-to-speech translation systems. In the second section two papers explore the use of speech technologies in language learning. The third section presents a work on language modeling used for speech recognition. The chapters in section Text-to-speech systems and emotional speech describe corpus-based speech synthesis and highlight the importance of speech prosody in speech recognition. In the fifth section the problem of speaker diarization is addressed. The last section presents various topics in speech technology applications like audio-visual speech recognition and lip reading systems.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821