

Addressing Morphological Variation in Alphabetic Languages

Paul McNamee
Human Language Technology
Center of Excellence
Johns Hopkins University
Baltimore, MD 21218, USA
paul.mcnamee@jhu.edu

Charles Nicholas
Dept. of Computer Science
and Electrical Engineering
UMBC
Baltimore, MD 21250, USA
nicholas@umbc.edu

James Mayfield
Human Language Technology
Center of Excellence
Johns Hopkins University
Baltimore, MD 21218, USA
james.mayfield@jhuapl.edu

ABSTRACT

The selection of indexing terms for representing documents is a key decision that limits how effective subsequent retrieval can be. Often stemming algorithms are used to normalize surface forms, and thereby address the problem of not finding documents that contain words related to query terms through inflectional or derivational morphology. However, rule-based stemmers are not available for every language and it is unclear which methods for coping with morphology are most effective. In this paper we investigate an assortment of techniques for representing text and compare these approaches using data sets in eighteen languages and five different writing systems.

We find character n-gram tokenization to be highly effective. In half of the languages examined n-grams outperform unnormalized words by more than 25%; in highly inflective languages relative improvements over 50% are obtained. In languages with less morphological richness the choice of tokenization is not as critical and rule-based stemming can be an attractive option, if available. We also conducted an experiment to uncover the source of n-gram power and a causal relationship between the morphological complexity of a language and n-gram effectiveness was demonstrated.

Categories and Subject Descriptors

H.3.1 [Information Systems]: Content Analysis and Indexing—*linguistic processing, indexing*; H.3.3 [Information Systems]: Information Search and Retrieval

General Terms

Experimentation

Keywords

Tokenization, Stemming, Morphology, Character N-grams, CLIR

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR '09, July 19–23, 2009, Boston, Massachusetts, USA.
Copyright 2009 ACM 978-1-60558-483-6/09/07 ...\$5.00.

1. INTRODUCTION

Failure to normalize morphologically related words (*e.g.*, *swimmer*, *swam*, *swimming*), can prevent matches in full-text retrieval. The conventional approach is to apply a stemming algorithm to each word to transform document representations from bags of surface forms to bags of stemmed forms. Stemming is an approximation to morpheme identification. It is usually considered to be a performance enhancing technique, despite the fact that all stemming algorithms suffer from errors of over- and/or under-conflation. Though they are sometimes difficult to distinguish from one another, three broad classes of morphological processes result in surface forms that impair effective retrieval: *inflection*, *derivation*, and *word formation*.

Inflectional morphemes add information to root morphemes such as number (*e.g.*, *dog/dog+s*; *fox/fox+es*) and gender (*e.g.*, *act+or/act+ress*, though English does not often inflect for gender). Other functions such as negation (*e.g.*, *un+happy*) and comparison (*e.g.*, *fast/fast+er/fast+est*) can be indicated with inflectional (or grammatical) morphemes, though sometimes these are expressed through function words (*e.g.*, *not happy*). The process of adding inflectional morphemes by attaching them to root morphemes is called agglutination. Some languages separate each morpheme into distinct words (*e.g.*, Chinese and Vietnamese), and these languages are termed *isolating*. However, affixation, the use of prefixes and suffixes to attach morphemes is extremely common. Languages that do this extensively are termed *agglutinative*. Languages vary in the degree of inflection and lie somewhere on the spectrum from isolating to strongly agglutinative. For example, English nouns only have two cases (singular and plural), but in Finnish, a highly agglutinative language, nouns can have fifteen different cases.

Derivational morphology transforms words from one syntactic class into another. For example *compute* (verb) can produce *computer* (noun); or *boy* (noun) can become an adjective through addition of the suffix *-ish*.

There are a variety of other methods for producing new words in a language, including:

- *foreign borrowing*: *umbrella* (Italian) becomes *umbrella* (English); *quiche* and *trompe l'oeil* are borrowed unmodified from French.
- *acronyms*: USA, NASA, MIT, and IBM are all derived from the initial letters in their corresponding words.
- *clipping*: compression of *professor* to *prof*, or *gymnasium* to *gym*.

- *blending*: fusion of component words into a shortened single form, such as *brunch* from *breakfast + lunch*.
- *compounding*: concatenation of two or more words to form a new word (e.g., *pickpocket*, *airport*, *airplane*, *girlfriend*, *mother-in-law*, *underachieve*). Like agglutination, compounding is more productive in some languages than others, and noun-noun compounding is a significant feature of Germanic languages.

Some of these processes are more harmful than others. Compounding, because it is so pervasive, is often given special treatment.

There are other linguistic phenomena that complicate information retrieval, including the classic problems of polysemy, where a word can have multiple meanings, and synonymy, where the same concept can be expressed with different word choices. Dialectical and spelling variants (and errors) have deleterious effect; however, the aim of this paper is to focus on issues of morphology in alphabetic languages. Towards that end we consider a variety of ways to represent text and compare them to each other while keeping other aspects of the process fixed.

The textual representations we investigated are described in Section 2. In Section 3 our data and methods are described and in Section 4 we present experimental results. Section 5 examines language variability and presents additional experiments that provide insight into why character n-grams are an effective technique. We summarize our findings in Section 6.

2. REPRESENTING TEXT

There are a number of operations in processing text, which while they are deserving of study, are so commonplace that they were adopted for all of the conditions examined. These include case folding, punctuation removal, and truncation of long numeric quantities to at most six digits. In this paper we examine 18 tokenization alternatives. Each is described below. Examples of indexing terms generated from the phrase *medical doctors* are given in Table 1 for each of the tokenization methods.

2.1 Words and Stemmed Forms

Our baseline condition, *words*, is formed from tokens delimited by spaces. Plain words are commonly used in web search where efficiency is paramount and precision is valued over recall. Words are also well justified in languages with little morphological complexity. Unlike stemming algorithms there are no errors of over-conflation, but words do suffer from polysemy.

Rule-based stemming is based on linguistically-inspired transformations. Snowball is a stemming algorithm compiler developed by Porter¹. Given a language-specific ruleset the compiler can produce source code that transforms surface forms into stems. For about half of the languages studied we were able to run Snowball (*snow*).

The motivation behind statistical stemmers is that they are language-neutral and thus universally applicable. We used the Morfessor algorithm² [4] (*morf*), which is designed to accommodate languages with concatenative morphology. The algorithm does not restrict the number of morphemes

¹Available at <http://snowball.tartarus.org/>

²Available at <http://www.cis.hut.fi/projects/morpho/>

Table 1: Examples of indexing term formation.

words	medical, doctors
snow	medic, doctor
morf	medical, doctor, s
devowel	m.d.c.l, d.ct.rs
soundex	5030204, 3023062
lfs4	edic, doct
lfs5	medic, docto
trun4	medi, doct
trun5	medic, docto
3-grams	_me, med, edi, dic, ica, cal, al_, Ld, _do, ...
4-grams	_med, medi, edic, dica, ical, cal_, al_d, Ldo, _doc, doct, ...
5-grams	_medi, medic, edica, dical, ical_, cal_d, al_do, Ldoc, _doct, ...
6-grams	_medic, medica, edical, dical_, ical_d, cal_do, al_doc, Ldoct, ...
7-grams	_medica, medical, edical_, dical_d, ical_do, cal_doc, al_doct, ...
sk41	regular 4-grams in addition to m.dic, me.ic, med.c, e.ica, ed.ca, edi.a, ..., a._do, al.do, al_o, l.doc, L.oc, L.d.c, ...
wisk41	word-internal 4-grams in addition to m.dic, me.ic, med.c, e.ica, ed.ca, edi.a, ...,
win4	_med, medi, edic, dica, ical, cal_, _doc, doct, octo, ctor, tors, ors_
win5	_medi, medic, edica, dical, ical_, _doct, docto, octor, ctors, tors_

that can be present in a single word and it possesses no explicit knowledge about a language; it only requires a list of words in the language. The algorithm is based on the minimum description length principle and it optimizes a cost function that measures how well the model represents the observed data and the combined lengths of the segments that make up the model's vocabulary. The output is a segmentation for each word. For example, *seabirds* is represented as *sea+bird+s*. In our experiments all of a word's segments were included in the inverted file.

2.2 Phonetic Transformations

Several phenomena create orthographic variants that could be normalized. Examples include dialect (e.g., *color / colour*), alternate transliterations (e.g., *Gorbachev / Gorbachyov*), spelling variation (e.g., *judgement / judgment*), and spelling mistakes (e.g., *congratulations / congradulations*). Additionally inflectional variation, particularly conjugation, is often achieved through vowel substitution (e.g., *write / wrote, throw / threw*).

We considered two forms of normalization, though the number of possibilities here is enormous. Vowel transformation (*devowel*) was accomplished by replacing all vowels with a unique symbol such as mapping 'a', 'e', etc. with '.' (dot), and by collapsing adjacent vowels to a single symbol. The other transformation was based on the Soundex algorithm (*soundex*) [14]. Unlike the traditional algorithm all letters were replaced with numerals, entire words were transformed (i.e., the resulting string was not limited to 4 characters), and characters outside the English alphabet were replaced with the digit '7'. This method was only applied to languages written in the Latin script.

2.3 Single Word Fragments

Selection of a single substring from a word has the potential to provide morphological normalization if the root morpheme can be identified. Approaches based on fixed-length prefixes [1] and substrings [17] have been proposed and both methods are considered here. Truncation of words to at most the first 4 (*trun4*) or 5 (*trun5*) characters is motivated by the observation that many languages rely heavily on suffixation. However, this approach is not likely to help much in the presence of prefixation and compounding. Frequency-based selection of a substring is based on the principle that affixes are likely to be common substrings, so corpus-informed selection of the least frequent substring of length 4 (*lfs4*) or 5 (*lfs5*) might coincide with the root morpheme.

2.4 Character N-grams

Fixed-length character n-grams have been noted for their language-independence and they have been used with good effect in European languages [19] and ideographic languages of Asia [2, 5, 22]. The technique provides significant redundancy in the representation of a word – the representation for a text is based on the number of letters it contains, not the number of words. This redundancy has the advantage of not requiring precise identification of root morphemes because a sliding window of length n will be sure to overlap morphemes. The main drawback of the approach is in increased disk space and run-time costs associated with significantly larger indexing representations.

Lengths of $n = 3$ to $n = 7$ were considered using word-spanning substrings, which may have a benefit in capturing some phrasal cues. Lengths of $n = 4$ and $n = 5$ have been reported to be the most effective [19]. Additionally word internal n-grams of lengths 4 & 5 were studied (*win4*, *win5*).

Character skipgrams, n-grams with skipped letters, have been proposed for fuzzy name matching [25], for normalization in Arabic [21], a language with root-and-template morphology, and as a means of finding translations in related languages. Järvelin *et al.* [11] examined skip bigrams (two characters with a single skip); however, multiple, and even non-adjacent skips are possible. Here 4-grams are combined with strings of length 5 that retain 4 letters and replace a single internal letter with a dot symbol. Skipgrams were created from both word-spanning (*sk41*) and word-internal (*wisk41*) n-grams.

3. METHODOLOGY

3.1 Data

Large multilingual collections from TREC³, CLEF⁴, and FIRE⁵ were used in our experiments. The test sets are generally comprised of newswire articles and information about the collections is given in Table 2.

Each evaluation typically created 50 or so queries per year and sometimes the document collections for a given language increased in size as additional corpora became available. Pooling was performed to efficiently create relevance judgments for the topic sets, however post-hoc use of these benchmarks for comparative evaluation is believed to be reliable.

³<http://trec.nist.gov/>

⁴<http://www.clef-campaign.org/>

⁵<http://www.isical.ac.in/~fire/>

Table 2: IR test collections for 18 languages.

	Language	Queries	Documents	Evaluation
AR	Arabic	75	383,872	TREC '01-'02
BG	Bulgarian	149	85,427	CLEF '05-'07
BN	Bengali	50	123,040	FIRE '08
CS	Czech	50	81,735	CLEF '07
DE	German	192	294,805	CLEF '00-'03
EN	English	367	87,653	CLEF '00-'07
ES	Spanish	156	454,041	CLEF '01-'03
FA	Farsi	50	166,774	CLEF '08
FI	Finnish	120	55,344	CLEF '02-'04
FR	French	333	177,450	CLEF '00-'06
HI	Hindi	45	95,213	FIRE '08
HU	Hungarian	148	49,530	CLEF '05-'07
IT	Italian	181	157,558	CLEF '00-'03
MR	Marathi	49	99,359	FIRE '08
NL	Dutch	156	190,605	CLEF '01-'03
PT	Portuguese	146	210,734	CLEF '04-'06
RU	Russian	62	16,715	CLEF '03-'04
SV	Swedish	102	142,819	CLEF '02-'03

3.2 Experimental Design

Queries were formed using *title* and *description* fields from the topic statements. As our focus is on comparing different tokenization techniques we wanted the experiment to reflect only changes in the indexing representation for documents (and queries). Therefore we elected not to employ relevance feedback in our experiments because it might conflate issues of expansion methods and term weighting with the selection of indexing terms.

Performance was measured using mean average precision (MAP) based on the number of queries shown in Table 2. Queries with no known relevant documents did not affect the calculation. Significance testing was performed with the paired *t*-test [3] using the available queries from multiple years to increase the sensitivity of the experiments.

3.3 Retrieval Model

A statistical language modeling approach [8, 20] was used for document ranking and smoothing was accomplished using linear interpolation using a constant of $\lambda = 0.5$ in all conditions:

$$P(D|Q) \propto \prod_{t \in Q} \lambda P(t|D) + (1 - \lambda) P(t|C) \quad (1)$$

Relative document term frequency was used to estimate $P(t|D)$. $P(t|C)$ was based on the mean relative document term frequency from documents in the collection.

4. RESULTS

Table 3 presents mean average precision for the indexing variants described in Section 2. Two averages across the languages are given. The first one listed is based on the eight languages supported by the Snowball stemmer.⁶ The second is based on the full set of 18 languages.

When only the Snowball languages are considered we observe that *snow* has an 11.5% relative advantage over unnormalized *words*. Morfessor segments and the least frequent

⁶Dutch, English, Finnish, French, German, Italian, Portuguese, and Spanish.

Table 3: Comparison of tokenization methods using MAP.

	<i>words</i>	<i>snow</i>	<i>morf</i>	<i>devowel</i>	<i>soundex</i>	<i>lfs4</i>	<i>lfs5</i>	<i>trun4</i>	<i>trun5</i>
AR	0.2054		0.2216	0.1973		0.2373	0.2267	0.1453	0.2148
BG	0.2164		0.2703	0.2136		0.2822	0.2442	0.2807	0.2959
BN	0.2630		0.2933	0.2664		0.2886	0.2692	0.2967	0.3058
CS	0.2270		0.3215	0.2556	0.2152	0.2567	0.2477	0.3039	0.3005
DE	0.3303	0.3695	0.3994	0.3170	0.2725	0.3464	0.3522	0.3202	0.3656
EN	0.4060	0.4373	0.4018	0.3843	0.2784	0.4176	0.4175	0.3900	0.4216
ES	0.4396	0.4846	0.4451	0.4263	0.3478	0.4485	0.4517	0.4249	0.4666
FA	0.3617		0.3559	0.3623		0.3629	0.3506	0.3505	0.3645
FI	0.3406	0.4296	0.4018	0.3368	0.2762	0.3995	0.4033	0.4060	0.4652
FR	0.3638	0.4019	0.3680	0.3516	0.2609	0.3882	0.3834	0.3541	0.3953
HI	0.2429		0.2477	0.3054		0.2484	0.2542	0.2732	0.2914
HU	0.1976		0.2921	0.2011	0.1564	0.2836	0.2668	0.2876	0.3082
IT	0.3749	0.4178	0.3474	0.3860	0.2889	0.3741	0.3673	0.3388	0.3963
MR	0.2572		0.3310	0.2779		0.2939	0.2626	0.3673	0.3477
NL	0.3813	0.4003	0.4053	0.3671	0.2963	0.3836	0.3846	0.3712	0.3946
PT	0.3162		0.3287	0.2956	0.2016	0.3418	0.3347	0.3144	0.3423
RU	0.2671		0.3307	0.2881		0.2875	0.3053	0.3216	0.3739
SV	0.3387	0.3756	0.3738	0.3279	0.2993	0.3638	0.3467	0.3209	0.3770
Average (<i>snow</i> langs)	0.3719	0.4146 11.5%	0.3928 5.6%	0.3621 -2.6%	0.2900 -22.0%	0.3902 4.9%	0.3883 4.4%	0.3658 -1.7%	0.4103 10.3%
Average	0.3072		0.3409 11.0%	0.3089 0.6%		0.3336 8.6%	0.3260 6.1%	0.3260 6.1%	0.3571 16.2%

<i>3-gram</i>	<i>4-gram</i>	<i>5-gram</i>	<i>6-gram</i>	<i>7-gram</i>	<i>sk41</i>	<i>wisk41</i>	<i>win4</i>	<i>win5</i>	
0.2516	0.2731	0.2356	0.2035	0.1699	0.2513	0.2522	0.2757	0.2390	AR
0.2271	0.3105	0.2820	0.2528	0.2161	0.2919	0.2945	0.3016	0.2866	BG
0.2826	0.3247	0.3173	0.2792	0.2400	0.3144	0.2821	0.3051	0.2770	BN
0.2792	0.3294	0.3223	0.2918	0.2536	0.3267	0.3261	0.3329	0.3245	CS
0.3188	0.4098	0.4201	0.3961	0.3632	0.4094	0.4039	0.4045	0.4129	DE
0.2588	0.3990	0.4152	0.3903	0.3556	0.4022	0.3894	0.3948	0.4037	EN
0.3010	0.4597	0.4609	0.4252	0.3621	0.4488	0.4582	0.4578	0.4662	ES
0.3032	0.3986	0.3821	0.3339	0.2870	0.3906	0.3333	0.3687	0.3300	FA
0.3591	0.4989	0.5078	0.4692	0.4323	0.4974	0.4867	0.5006	0.4882	FI
0.2544	0.3844	0.3930	0.3660	0.3201	0.3856	0.3844	0.3796	0.3886	FR
0.2448	0.3305	0.3271	0.2932	0.2517	0.3243	0.2694	0.2886	0.2556	HI
0.2778	0.3746	0.3624	0.3335	0.3030	0.3693	0.3577	0.3714	0.3490	HU
0.2177	0.3738	0.3997	0.3669	0.3217	0.3817	0.3833	0.3672	0.4053	IT
0.3886	0.4114	0.3739	0.3168	0.2680	0.3740	0.3751	0.4164	0.3715	MR
0.3326	0.4219	0.4243	0.3960	0.3663	0.4222	0.4040	0.4141	0.4050	NL
0.2213	0.3358	0.3524	0.3223	0.2834	0.3428	0.3359	0.3367	0.3475	PT
0.3252	0.3406	0.3330	0.3181	0.3028	0.3346	0.3399	0.3610	0.3393	RU
0.3244	0.4236	0.4271	0.4004	0.3713	0.4142	0.4119	0.4126	0.4234	SV
0.2959 -20.4%	0.4214 13.3%	0.4310 15.9%	0.4013 7.9%	0.3616 -2.8%	0.4202 13.0%	0.4152 11.6%	0.4164 12.0%	0.4242 14.1%	Average (<i>snow</i> langs)
0.2871 -6.5%	0.3778 23.0%	0.3742 21.8%	0.3420 11.3%	0.3038 -1.1%	0.3712 20.8%	0.3604 17.3%	0.3716 21.0%	0.3619 17.8%	Average

Table 4: Performance relative to *words* baseline.

	<i>words</i>	<i>snow</i>		<i>trun5</i>		<i>4-grams</i>		<i>5-grams</i>		Top method	
AR	0.2054			0.2148	+4.6%	0.2731 [▲]	+33.0%	0.2356 [△]	+14.7%	0.2731 [▲]	+33.0%
BG	0.2164			0.2959 [▲]	+36.7%	0.3105 [▲]	+43.5%	0.2820 [▲]	+30.3%	0.3105 [▲]	+43.5%
BN	0.2630			0.3058 [△]	+16.3%	0.3247 [△]	+23.5%	0.3173 [△]	+20.6%	0.3247 [△]	+23.5%
CS	0.2270			0.3005 [▲]	+32.4%	0.3294 [▲]	+45.1%	0.3223 [▲]	+42.0%	0.3329 [▲]	+46.7%
DE	0.3303	0.3695 [▲]	+11.9%	0.3656 [▲]	+10.7%	0.4098 [▲]	+24.1%	0.4201 [▲]	+27.2%	0.4201 [▲]	+27.2%
EN	0.4060	0.4373 [▲]	+7.7%	0.4216 [△]	+3.8%	0.3990	-1.7%	0.4152	+2.3%	0.4373 [▲]	+7.7%
ES	0.4396	0.4846 [▲]	+10.2%	0.4666 [△]	+6.1%	0.4597	+4.6%	0.4609 [△]	+4.8%	0.4846 [▲]	+10.2%
FA	0.3617			0.3645	+0.8%	0.3986 [△]	+10.2%	0.3821	+5.6%	0.3986 [△]	+10.2%
FI	0.3406	0.4296 [▲]	+26.1%	0.4652 [▲]	+36.6%	0.4989 [▲]	+46.5%	0.5078 [▲]	+49.1%	0.5078 [▲]	+49.1%
FR	0.3638	0.4019 [▲]	+10.5%	0.3953 [▲]	+8.7%	0.3844 [△]	+5.7%	0.3930 [▲]	+8.0%	0.4019 [▲]	+10.5%
HI	0.2429			0.2914 [▲]	+20.0%	0.3305 [▲]	+36.1%	0.3271 [▲]	+34.7%	0.3305 [▲]	+36.1%
HU	0.1976			0.3082 [▲]	+56.0%	0.3746 [▲]	+89.6%	0.3624 [▲]	+83.4%	0.3746 [▲]	+89.6%
IT	0.3749	0.4178 [▲]	+11.4%	0.3963	+5.7%	0.3738	-0.3%	0.3997 [△]	+6.6%	0.4178 [▲]	+11.4%
MR	0.2572			0.3477 [▲]	+35.2%	0.4114 [▲]	+60.0%	0.3739 [▲]	+45.4%	0.4164 [▲]	+61.8%
NL	0.3813	0.4003 [△]	+5.0%	0.3946	+3.5%	0.4219 [▲]	+10.6%	0.4243 [▲]	+11.3%	0.4243 [▲]	+11.3%
PT	0.3162			0.3423 [△]	+8.3%	0.3358	+6.2%	0.3524 [▲]	+11.4%	0.3524 [▲]	+11.4%
RU	0.2671			0.3739 [▲]	+40.0%	0.3406 [▲]	+27.5%	0.3330 [△]	+24.7%	0.3739 [▲]	+40.0%
SV	0.3387	0.3756 [▲]	+10.9%	0.3770 [△]	+11.3%	0.4236 [▲]	+25.1%	0.4271 [▲]	+26.1%	0.4271 [▲]	+26.1%

substring methods have modest gains of about 5%, but the n-grams of lengths $n = 4, 5$ score the highest. The *5-grams* at 15.9% edge out the *4-grams* at 13.3%.

Interestingly a seemingly disruptive transformation such as discarding information about specific vowels (*devowel*) causes little harm (-2.6%), but the modified *soundex* drops performance markedly (-22.0%).

When all of the languages are examined direct comparisons to Snowball cannot be made, but we see the 4- and 5-grams grow to a 22-23% advantage over words. 5-grams attain higher MAP in all 18 languages compared to plain words. 4-grams are marginally higher on average, but score lower than words in English and Italian. The word-spanning n-grams perform slightly better than the word-internal and skipgram forms. 3-grams are clearly too short to be effective, but the longer lengths of $n = 6$ and $n = 7$ are still improvements over words.

The best non n-gram approach is *trun5* (+16.2%), which retains at most the first five letters of a word. This gives two-thirds of the benefit of 4-grams but requires storage of only one posting entry per word. Performance is only slightly behind Snowball, but the method improves on words for all 18 languages.

In Table 4 we examine the top performing methods in greater detail. Mean average precision is given along with the relative improvement over the *words* baseline. Significant improvements with $p < 0.01$ are indicated with solid triangles (▲); open triangles indicate improvements with $p < 0.05$ (△). Gains with 5-grams are statistically significant in 16/18 cases; 4-grams and *trun5* each lead to significant improvements for 14 of the 18 languages.

Differences between n-grams and Snowball tend to be significant. 4-grams and 5-grams are statistically better in German, Finnish, and Swedish. Snowball is significantly better in English and Spanish (both), and in French and Italian (4-grams).

In Table 5 disk space usage and mean query response times are given for each of the the methods in Table 4. N-gram indexing can consume 6 times as much storage and queries can take 8 times as long to execute.

Table 5: Storage requirements and execution speed using the CLEF 2003 English collection.

	Dict (MB)		Inv file (MB)		Query (sec)	
<i>words</i>	4.7		60.0		0.51	
<i>snow</i>	3.5	-26%	50.4	-16%	0.65	+27%
<i>trun5</i>	1.6	-66%	47.0	-22%	0.90	+77%
<i>4-grams</i>	1.8	-62%	232	+287%	4.08	+700%
<i>5-grams</i>	10.9	+131%	391	+552%	4.42	+767%

5. DISCUSSION

5.1 Language Variation

The results on the Hungarian and Marathi collections are of particular note: n-grams were a better choice than words for $3 \leq n \leq 7$. In Hungarian 4- and 5-grams were 80% more effective than words; for Marathi 4-grams yielded a 60% improvement.

N-grams were able to provide at least a 25% relative improvement in Arabic, Bulgarian, Czech, German, Finnish, Hindi, Hungarian, Marathi, Russian, and Swedish. Notably absent from this list are any of the Romance languages. In fact, n-grams (*e.g.*, 5-grams) have the least advantage in English and in French, Italian, and Spanish.

Linguistic typology appears to affect the success of n-gram tokenization. One hypothesis that would account for this is that n-gram effectiveness is tied to morphological complexity. Though such methods are not without controversy among linguists, there have been studies that attempted to quantify morphological complexity using principles from information theory.

Juola [12] examined translations of the Bible and erased morphology from each in the following way. Each word (or type) in a text is replaced with a unique symbol, a randomly selected integer. After this has been done to the entire text the words that normally exhibit morphological regularity, such as *jump*, *jumped*, *jumping*, no longer bear an obvious relationship to one another any more than do the numbers 18, 5429, and 1641. Juola then compared languages based

Table 6: 5-gram effectiveness and linguistic complexity in European languages.

	Word Length	Juola Ratio	Kettunen Ratio	5-gram Gain
HU	5.99		1.1421	83.40%
FI	7.23	1.1253	1.1637	49.09%
CS	5.38		1.0867	41.98%
BG	5.02			30.31%
DE	5.98		1.1660	27.19%
SV	5.26		1.1252	26.10%
RU	5.93	1.0456		24.67%
PT	4.89		1.0676	11.45%
NL	5.17	0.9949	1.1189	11.28%
FR	4.79	1.0117	1.0622	8.03%
IT	5.08		1.0518	6.62%
ES	4.89		1.0624	4.85%
EN	4.68	0.9717	1.0529	2.27%
ρ	0.7771	0.9054	0.6761	

on the ratio of the compressibility of the original text to the compressibility of the morphologically degraded text; the program *gzip* was used as a way of approximating the Kolmogorov complexity of the texts. Kettunen *et al.* [13] followed the approach described by Juola and performed a similar analysis using translations of the European Union Constitution in 21 languages and the program *bzip2*.

In Table 6 data is presented that shows for each language: (1) the mean word length, by token; (2) the ratio that Juola computed to indicate morphological complexity (larger indicates greater complexity), if available; (3) Kettunen *et al.*'s corresponding ratio for the language, if available⁷; and (4) the relative improvement observed with 5-gram tokenization. The three estimates of morphological complexity can be used to rank languages by inferred complexity. Similarly the relative gains attained using 5-grams instead of words can also be used to order languages from those that gain much (*e.g.*, Hungarian and Finnish) down to those that gain little (*e.g.*, English and Spanish). The table also gives Spearman rank correlation coefficients, which show moderate to large correlations between each of the three estimates of morphological complexity and the gains attainable with 5-grams.

5.2 Reasons for N-gram Effectiveness

There are a number of factors that could be the underlying cause of the 20+% improvement with n-grams. The gains observed with n-grams could be due to:

- robustly coping with spelling variations (*e.g.*, *Jacobson/Jacobssen* or *color/colour*) because of the redundancy that comes from generating multiple indexing terms from different sections of a word;
- the word-spanning n-grams that provide evidence about word adjacency;
- or, handling morphological variation, including inflectional changes and compounding.

Spelling normalization can certainly provide gains, however this is somewhat difficult to quantify. We can say pretty con-

⁷Kettunen *et al.*'s ratios tend to be slightly higher, but for languages in common the agreement in rankings is good.

Table 7: Example permutations using words from the CLEF 2000 English corpus.

Original Word	DF	Shuffled Form	DF
ate	613	aet	1316
eat	2459	tae	2459
tea	741	aet	1316
team	16605	tema	16605
meat	1217	maet	1217
luau	20	luaa	20
lull	119	lull	119
golfer	258	legfro	258
golfed	5	dofegl	5
golfing	97	ligfgon	97
golfball	2	gaboflll	2

fidently that misspellings alone would not explain the large gains observed – spelling errors and variations are just not that common. Word-spanning and word-internal n-grams can be directly compared. From Table 3 we see that *4-grams* and *win4* achieve very similar performance. While word-spanning 5-grams are a bit more effective than their word-internal counterparts, it appears that the limited phrasal information from word-crossing n-grams would not explain what is occurring.

In an effort to establish whether or not coping with morphological processes such as inflection, derivation, and compounding is the prime reason behind n-gram's monolingual effectiveness, we can attempt to remove morphology from language and see what changes occur. Inspired by Juola's work in degrading morphology [12] a method of altering every word in the lexicon will be performed and retrieval experiments can be run against indexes created using word-based or n-gram-based tokenization on the transformed words. If the relative advantage of character n-grams disappears this will be support that it is by addressing morphology that n-grams improve on word-based indexing.

To remove morphological regularity we randomly shuffle the order of the characters in each word. Each word is thus transformed in a way that preserves its length, but makes it very hard to observe similarity between related lexemes (see Table 7). Short words and those with many repeated characters (*e.g.*, *lull*) will bear a strong resemblance to their original forms even after scrambling the letters, but the majority of surface forms will be considerably transformed. The effect on word-based indexing should be minimal, although some increase in polysemy is possible due to manufactured connotations in the transformed representations. This might happen because anagrams, such as *team* and *meat*, could become cognates through shuffling if each was converted to *eamt*. This method of removing morphology will not distinguish between morphological processes such as inflection and compounding; some types of morphology may have a more significant impact on retrieval than others, but this experiment will not explain the relative contribution of different morphological processes in a language.

The first group in Table 7 illustrates that additional connotations will occur as both *ate* and *tea* are transformed to *aet*, which has a document frequency near the sum of the number of documents that the original terms appeared in. Anagrams *team* and *meat* remain separate in the transformed space. No constraint was imposed to ensure that shuffled

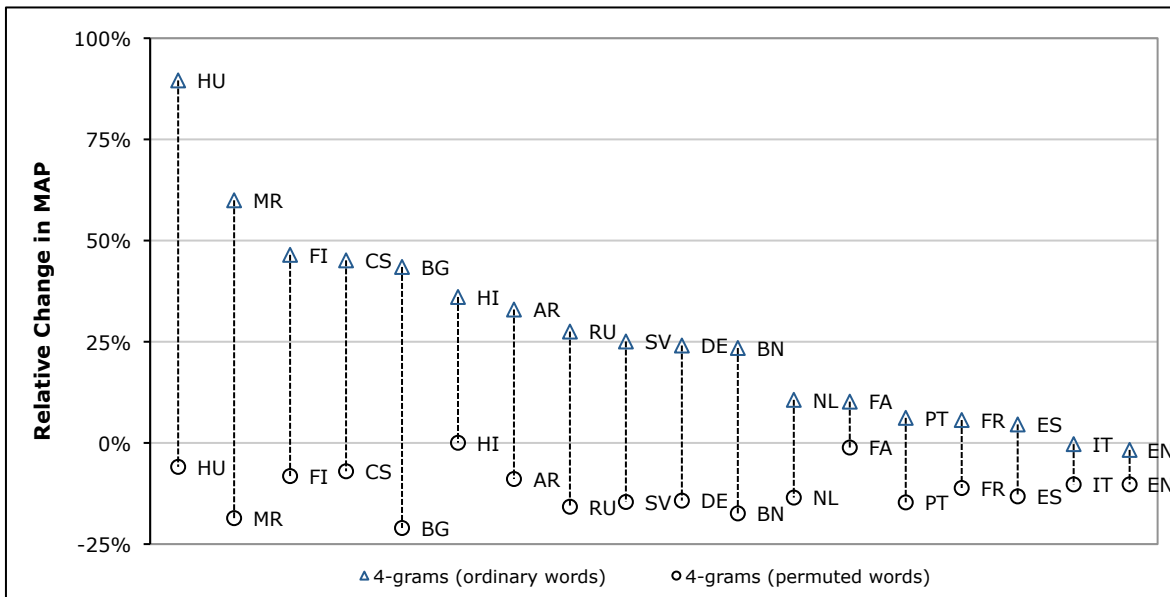


Figure 1: Comparative efficacy of 4-grams against words, when the order of letters in words has, and has not, been scrambled throughout the corpus.

forms differed from their original strings; while this is unlikely with longer terms, the word *lull*, which only has four possible forms depending on where the letter ‘u’ is positioned, is an example of a word left unaltered. Finally, the last grouping in the table demonstrates how related forms of the lexeme *golf* lack any resemblance in their scrambled representations.

We examined whether the relative effectiveness of n-grams changes when the letters of words are randomly scrambled. Figure 1 plots the percent change in performance for each language when character 4-grams are used instead of ordinary words. The vertical axis measures the relative gain (or loss) compared to words that is attributable to 4-gram indexing. The 0% threshold marks parity between the n-grams and words. Languages are ordered left-to-right by decreasing n-gram performance. The triangles indicate conventionally produced 4-grams and circles are used for 4-grams that are generated from documents with permuted words.

We found that:

- No change occurs when using space-separated words as indexing terms⁸.
- N-grams of lengths $n = 4$ and $n = 5$ perform markedly worse, suffering a 28% decline in mean average precision, averaged over all languages. Performance falls appreciably below that of word-based indexing. This decrease below 0% makes sense since n-grams are a conflationary technique and we have prevented the more desirable conflations.

These results give strong evidence that it is the ability of overlapping character n-grams to capture regularity across morphologically related words forms that gives them their primary advantage.

⁸This is not visible in Figure 1, but for *words* MAP was unaffected by the letter scrambling.

If it is the isolation of the root morpheme (or in compounds, roots) that is key, then these findings also suggest why longer length n-grams such as $n = 6$ and $n = 7$ are less effective than $n = 4$ and $n = 5$: longer sequences of characters are not focused on morphemes and fail to match some inflected allomorphs.

This also gives hope that the computational expense incurred with n-gram indexing can be reduced through aggressive pruning based on detecting morphological roots.

5.3 Related Work

On early English collections (e.g., Cranfield, Medlars, and CACM) Harman found little advantage in stemming [7]. Hull reported average improvements of 1 to 3% [10]; however, Krovetz found larger differences using CACM, NPL, TIME, and WEST [15] which he ascribed to addressing derivational morphology. These foundational studies were based on the only available test collections of the time, which were in English. Since the number of English inflectional forms is low, it is not surprising that the observed differences were not large.

The CLEF data sets have facilitated investigation of tokenization methods in multiple European languages. Hollink *et al.* compared words, stems, lemmas, and some combinations such as n-grams over stems and lemmas [9]. Their results are consistent with those in this study. In particular they found that stemming worked well in Romance languages and for only one of eight languages did they find a technique that outperformed 4-grams significantly. More recently, Savoy has shown ‘light stemmers’, ones that only attempt to remove inflectional affixes, can be very effective [23]. He reported relative gains from 8% to 42%.

Corpus-based approaches to stemming have been studied for over 30 years, beginning with methods based on successor varieties [6]. Xu and Croft examined word co-occurrence statistics to enhance an existing stemmer or to induce one[24].

Recently the Morpho Challenge competition investigated the use of unsupervised morphological analysis for information retrieval, using English, Finnish, and German test sets from CLEF. Nearly all of the analyses produced by competing systems outperformed the baseline condition, which left surface forms unaltered [16].

McNamee has conducted related tokenization experiments in European languages as well as experiments into full-text character skip-gram indexing and the use of automated relevance feedback with n-grams [18].

6. CONCLUSIONS

We compared a number of tokenization variants on test sets in diverse languages. The top performing method was word-spanning character n-gram indexing using $n = 4$ or $n = 5$, which is consistent with earlier reports in the literature. Other n-gram variants performed well, but did not do as well as the nearly identically performing 4-grams and 5-grams. Another top performer was truncation of words to at most 5 letters. This method achieved much of the benefit of the n-grams yet it does not incur any run-time or disk space disadvantage. Both of these approaches are language-independent.

We noted differences based on language family. Rule-based stemming using the Snowball rulesets performed well in English and the Romance family, and in those languages it tended to outshine n-grams. In highly complex languages it proved essential to control for morphology to obtain the best results. In several languages relative improvements of 40% to 80% compared to words could be obtained using n-grams. We also showed strong correlations between 5-gram effectiveness and language complexity and conducted an experiment that showed that n-grams lose their power when morphology is removed from a language.

With the recent availability of test collections in diverse language families, we can now say that controlling for morphology is not optional, but is vital for effective multilingual IR. Furthermore, n-gram indexing is a strong default method that other approaches should be measured against.

7. REFERENCES

- [1] P. Ahlgren and J. Kekäläinen. Indexing strategies for Swedish full text retrieval under different user scenarios. *Information Processing and Management*, 43(1):81–102, 2007.
- [2] A. Chen, J. He, L. Xu, Gey, F. C., and J. Meggs. Chinese text retrieval without using a dictionary. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49, 1997.
- [3] G. V. Cormack and T. R. Lynam. Validity and power of t-test for comparing MAP and GMAP. In *Proceedings of ACM SIGIR*, pages 753–754, 2007.
- [4] M. Creutz and K. Lagus. Unsupervised discovery of morphemes. In *ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30, 2002.
- [5] S. Foo and H. Li. Chinese word segmentation and its effect on information retrieval. *Information Processing and Management*, 40(1):161 – 190, 2004.
- [6] M. A. Hafer and S. F. Weiss. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10(11/12):371–385, 1974.
- [7] D. Harman. How effective is stemming? *JASIS*, 42(1):7–15, 1991.
- [8] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, 2001.
- [9] V. Hollink, J. Kamps, C. Monz, and M. de Rijke. Monolingual document retrieval for European languages. *Information Retrieval*, 7(1-2):33–52, 2004.
- [10] D. A. Hull. Stemming algorithms: A case study for detailed evaluation. *JASIS*, 47(1):70–84, 1996.
- [11] A. Järvelin, A. Järvelin, and K. Järvelin. S-grams: Defining generalized n-grams for information retrieval. *Information Processing and Management*, 43(4):1005–1019, 2007.
- [12] P. Juola. Measuring linguistic complexity: the morphological tier. *Journal of Quantitative Linguistics*, 5(3):206–213, 1998.
- [13] K. Kettunen, M. Sadeniemi, T. Lindh-Knuutila, and T. Honkela. Analysis of EU languages through text compression. In *FinTAL*, pages 99–109, 2006.
- [14] D. E. Knuth. *Art of Computer Programming, Volume 3: Sorting and Searching (2nd Edition)*. Addison-Wesley Professional, April 1998.
- [15] R. Krovetz. Viewing morphology as an inference process. In *ACM SIGIR 1993*, pages 191–202, 1993.
- [16] M. Kurimo, M. Creutz, and V. Turunen. Overview of Morpho Challenge in CLEF 2007. In *Working Notes of the CLEF 2007 Workshop*, 2007.
- [17] J. Mayfield and P. McNamee. Single n-gram stemming. In *Proceedings of ACM SIGIR*, pages 415–416, 2003.
- [18] P. McNamee. *Textual Representations for Corpus-Based Bilingual Retrieval*. PhD thesis, University of Maryland Baltimore County, Baltimore, MD, 2008.
- [19] P. McNamee and J. Mayfield. Character n-gram tokenization for European language text retrieval. *Information Retrieval*, 7(1-2):73–97, 2004.
- [20] D. R. H. Miller, T. Leek, and R. M. Schwartz. A hidden Markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 214–221, New York, NY, USA, 1999. ACM.
- [21] S. H. Mustafa. Character contiguity in n-gram based word matching: the case for Arabic text searching. *Information Processing and Management*, 41:819–827, 2004.
- [22] Y. Ogawa and T. Matsuda. Overlapping statistical word indexing: A new indexing method for Japanese text. In *SIGIR*, pages 226–234. ACM, 1997.
- [23] J. Savoy. Light stemming approaches for the French, Portuguese, German and Hungarian languages. In *SAC '06: Proceedings of the 2006 ACM symposium on applied computing*, pages 1031–1035, New York, NY, USA, 2006. ACM.
- [24] J. Xu and W. B. Croft. Corpus-based stemming using cooccurrence of word variants. *ACM Trans. Inf. Syst.*, 16(1):61–81, 1998.
- [25] J. Zobel and P. Dart. Finding approximate matches in large lexicons. *Software - Practice and Experience*, 25(3):331–345, 1995.