Benefits of Resource-Based Stemming in Hungarian Information Retrieval

Péter Halácsy[1] and Viktor Trón[2]

[1] Budapest University of Technology and Economics
Centre for Media Research
hp@mokk.bme.hu
[2] International Graduate College
Saarland University and University of Edinburgh
v.tron@ed.ac.uk

Abstract. This paper discusses the impact of resource-driven stemming in information retrieval tasks. We conducted experiments in order to identify the relative benefit of various stemming strategies in a language with highly complex morphology. The results reveal the importance of various aspects of stemming in enhancing system performance in the IR task of the CLEF ad-hoc monolingual Hungarian track.

The first Hungarian test collection for information retrieval (IR) appeared in the 2005 CLEF ad-hoc task monolingual track. Prior to that no experiments had been published that measured the effect of Hungarian language-specific knowledge on retrieval performance.

Hungarian is a language with highly complex morphology.[1] Its inventory of morphological processes include both affixation (prefix and suffix) and compounding. Morphological processes are standardly viewed as providing the grammatical means for (i) creating new lexemes (derivation) as well as (ii) expressing morphosyntactic variants belonging to a lexeme (inflection). To illustrate the complexity of Hungarian morphology, we mention that a nominal stem can be followed by 7 types of Possessive, 3 Plural, 3 Anaphoric Possessive and 17 Case suffixes yielding as many as 1134 possible inflected forms.

Similarly to German and Finnish, compounding is very productive in Hungarian. Almost any two (or more) nominals next to each other can form a compound (e.g., *üveg+ház+hat-ás* = glass+house+effect 'greenhouse effect'). The complexity of Hungarian and the problems it creates for IR is detailed in [1].

1 Stemming and Hungarian

All of the top five systems of the 2005 track (Table 1) had some method for handling the rich morphology of Hungarian: either words were tokenized to n-grams or an algorithmic stemmer was used.

[1] A more detailed descriptive grammar of Hungarian is available at http://mokk.bme.hu/resources/ir

C. Peters et al. (Eds.): CLEF 2006, LNCS 4730, pp. 99–106, 2007.

Table 1. The top five runs for Hungarian ad hoc monolingual task of CLEF 2005

part	run	map	stemming method
jhu/apl	aplmohud	41.12%	4gram
unine	UniNEhu3	38.89%	Savoy's stemmer + decompounding
miracle	xNP01ST1	35.20%	Savoy's stemmer
humminngbird	humHU05tde	33.09%	Savoy's stemmer + 4gram
hildesheim	UHIHU2	32.64%	5gram

The best result was achieved by JHU/APL with an IR system based on lanugage modelling in the run called `aplmohud` [2]. This system used a character 4-gram based tokenization. Such n-gram techniques can efficiently get round the problem of rich agglutinative morphology and compounding. For example the word *atomenergia* = 'atomic energy' in the query is tokenized to *atom, tome, omen, mene, ener, nerg, ergi, rgia* strings. When the text only contains the form *atomenergiával* = 'with atomic energy', the system still finds the relevant document.

Although this system used the Snowball stemmer together with the n-gram tokenization for the English and French tasks, the Hungarian results were nearly as good: English 43.46%, French 41.22% and Hungarian 41.12%. From these results it seems that the difference between the isolating and agglutinating languages can be eliminated by character n-gram methods.

Unine [3], Miracle [4] and Hummingbird [5] all employ the same algorithmic stemmer for Hungarian that removes the nominal suffixes corresponding to the different cases, the possessive and plural (`http://www.unine.ch/info/clef`).

`UniNEhu3` [3] also uses a language independent decompounding algorithm that tries to segment words according to corpus statistics calculated from the document collection [6]. The idea is to find a segmentation that maximizes the probability of hypothesized segments given the document and the language. Given the density of short words in the language, spurious segmentations can be avoided by setting a minimum length limit (8-characters in the case of Savoy) on the words to be segmented.

Among the 2005 CLEF contestants, [7] is especially important for us, since she uses the same baseline system. She developed four stemmers which implement successively more agressive stripping strategies. The lightest only strips some of the frequent case suffixes and the plural and the heaviest strips all major inflections. We conjecture that the order in which the suffix list is enriched is based on an intuitive scale of suffix transparency or meaningfulness which is assumed to impact on the retrieval results. In line with previous findings, she reports that the stemming enhances retrieval, with the most agressive strategy winning. However, the title of her paper 'Four stemmers and a funeral' aptly captures her main finding that even the best of her stemmers performs the same as a 6-gram method.

2 The Experimental Setting

In order to measure the effects of various flavours of stemming on retrieval performance, we put together an unsophisticated IR system. We used JAKARTA LUCENE 2.0, an off-the-shelf system to perform indexing and retrieval with ranking based on its default vector space model. No further ranking heuristics or post-retrieval query expansion is applied. Before indexing the XML documents are converted to plain text format, header (title, lead, source, etc.) and body sequentially appended. For tokenization we used Lucene's `LetterTokenizer` class: this considers every non-alphanumeric character (according to the Java `Character` class) as token boundary. All tokens are lowercased but not stripped of accents. The document and query texts are tokenized the same way including topic and description fields used in the search. Our various runs differ only in how these text tokens are mapped on terms for document indexing and querying. In the current context, we define stemming as solutions to this mapping. Tokens that exactly matched a stopword[2] before or after the application of our stemming algorithms were eliminated. Stemming can map initial document and query tokens onto zero, one or more (zero only for stopwords) terms. If stemming yields more than one term, each resulting term (but not the original token) was used as an index term. For the construction of the query each term resulting from the stemmed token (including multiple ones) was used as a disjuntive query term.

Using this simple framework allows us to isolate and compare the impact of various strategies of stemming. Using an unsophisticated architecture has the further advantage that any results reminiscent of a competitive outcome will suggest the beneficial effect of the particular stemming method even if no direct comparison to other systems is available due to different retrieval and ranking solution employed.

2.1 Strategies of Stemming

Instead of some algorithmic stemmers employed in all previous work, we leveraged our word-analysis technology designed for generic NLP tasks including morphological analysis and POS-tagging. `Hunmorph` is a word-analysis toolkit, with a language independent analyser using a language-specific lexicon and morphological grammar [8]. The core engine for recognition and analysis can (i) perform guessing (i.e., hypothesize new stems) and (ii) analyse compounds.

Guessing means that possible analyses (morphosyntactic tag and hypothesized lemma) are given even if the input word's stem is not in the lexicon. This feature allows for a stemming mode very similar to resourceless algorithmic stemmers if no lexicon is used. However, guessing can be used in addition to the lexicon.

To facilitate resource sharing and to enable systematic task-dependent optimizations from a central lexical knowledge base, the toolkit offers a general framework for describing the lexicon and morphology of any language. `hunmorph` uses the Hungarian lexical database and morphological grammar called

[2] We used the same stopword list as [7], which is downloadable at `http://ilps.science.uva.nl/Resources/HungarianStemmer/`

`morphdb.hu` [9]. `morphdb.hu` is by far the broadest-coverage resource for Hungarian reaching about 93% recall on the 700M word Hungarian Webcorpus [10]. We set out to test the impact of using this lexical resource and grammar in an IR task. In particular we wanted to test to what extent guessing can compensate for the lack of the lexicon and what types of affixes should be recognized (stripped). We also compared this technology with the two other stemmers mentioned above, Savoy's stemmer and Snowball.

Decompounding is done based on the compound analyses of `hunmorph` according to compounding rules in the resource. In our resource, only two nominals can form a compound. Although compounding can be used with guessing, this only makes sense if the head of the compound can be hypothesized independently, i.e., if we use a lexicon. We wanted to test to what extent compounding boosts IR efficiency.

Due to extensive morphological ambiguities, `hunmorph` often gives several alternative analyses. Due to limitations of our simple IR system we choose only one candidate. We have various strategies as to which of these alternatives should be retained as the index term. We can use (i) basic heuristics to choose from the alternants, or (ii) use a POS-tagger that disambiguates the analysis based on textual context.

As a general heuristics used in (i) and (ii) we prefer analyses that are neither compound nor guessed, if no such analysis exists then we prefer non-guessed compounds over guessed analyses.

(ii) involves a linguistically sophisticated method, pos-tagging that restricts the candidate set by choosing an inflectional category based on contextual disambiguation. We used a statistical POS-tagger [11] and tested its impact on IR performance. This method relies on the availability of a large tagged training corpus; if a language has such a corpus, lexical resources for stemming are very likely to exist. Therefore it seemed somewhat irrelevant to test the effect of pos-tagging without lexical resources (only on guesser output).[3]

If there are still multiple analyses either with or without POS-tagging, we found that choosing the shortest lemma for guessed analyses (agressive stemming) and the longest lemma for known analyses (blocking of further analysis by known lexemes) works best. When a compound analysis is chosen, lemmata of all constituents are retained as index terms.

3 Evaluation

Table 2 compares the performance of the various stemming algorithms. The clearest conclusion is the robust increase in precision achieved when stemming is used. Either of the three stemmers, Savoy, Snowball and Hunmorph is able to boost precision with at least 50% in both years. Pairwise comparisons with a paired t-test on topic-wise map values show that all three stemmers are significantly better than the baseline. Snowball and Savoy are not different. Most

[3] POS-tagging also needs sentence boundary detection. For this we trained a maximum entropy model on the Hungarian Webcorpus [12].

Table 2. Results of different stemming methods

stemmer	year	MAP	P10	ret/rel
Hunmorph	2005	0.3951	0.3900	883/939
lexicon+decomp	2006	0.3443	0.4320	1149/1308
Hunmorph	2005	0.3532	0.3560	836/939
lexicon	2006	0.2989	0.3840	1086/1308
Hunmorph	2005	0.3444	0.3400	819/939
no lexicon	2006	0.2785	0.3580	1025/1308
Snowball	2005	0.3371	0.3360	818/939
(no lexicon)	2006	0.2790	0.3820	965/1308
Savoy's	2005	0.3335	0.3360	819/939
(no lexicon)	2006	0.2792	0.3860	964/1308
baseline	2005	0.2153	0.2400	656/939
no stemmer	2006	0.1853	0.2700	757/1308

variants of Hunmorph also perform in the same range when it is used without a lexicon. In fact a comparison of different variants of this lexicon-free stemmer is instructive (see Table 3). Allowing for the recognition of derivational as well as inflectional suffixes (heavy) is always better than using only inflections (light). This is in line with or an extension to [7]'s finding that the more agressive the stemming the better. It is noteworthy that stemming verbal suffixation on top of nominal can give worse results (as it does for 2005). It is uncertain whether this is because of an increased number of false noun-verb conflations. Pairwise comparisons between these variants and the two other stemmers show no significant difference.

Table 3. Comparison of different lexicon-free stemmers. nom-heavy: nominal suffixes, all-heavy: nominal and verbal suffixes; nom-light: nominal inflection only, all-light: nominal and verbal inflection only.

guesser	year	MAP	P10	ret/rel
nom-heavy	2005	**0.3444**	**0.3400**	**819/939**
	2006	0.2785	0.3580	**1025/1308**
all-heavy	2005	0.3385	0.3520	817/939
	2006	**0.2837**	**0.3660**	1019/1308
nom-light	2005	0.3244	0.3360	808/939
	2006	0.2653	0.3540	962/1308
all-light	2005	0.3158	0.3240	811/939
	2006	0.2663	0.3560	1000/1308

Note again, that Hunmorph uses a morphological grammar to guess alteernative stems. This has clearly beneficial consequences because its level of sophistication allows us to exclude erroneous suffix combinations that Snowball (maybe

mistakenly) identifies such as *alakot→al* instead of *alak* and which cause faulty conflations. On the other hand, Hunmorph allows all the various possibilities that may occur in the language with a lexicon however phonotactically implausible they are. The simple candidate selection heuristics does not take this into account and may choose (randomly) the implausible *ábrákat→ábrá* instead of *ábra* (the base form) and thereby missing crucial conflations.

One of the most important findings is that using a lexicon is not significantly better than any of the resourceless alternatives when compounding is not used.

As expected from the previous discussion, the best results are achieved when decompounding is used. Our method of decompounding ensures that component stems are (at least potentially) linguistically correct, i.e., composed of real existing stems with the right category matching a compound pattern. This also implies that only the head category (right constituent) can be inflected on a lexical data-base, though the other components can also be derived. Using this method of decompounding has a clearly beneficial effect on retrieval performance increasing accuracy from 34%/30% up to 39%/34% on the 2005 and the 2006 collection respectively (both years show that compounding is significantly better than all alternatives not using compounding). The 2005 result of 39% positions our system at the second place beating Savoy's system which uses the alternative compounding. A more direct comparison would require changing only the compounding algorithm while keeping the rest of our system unchanged. Since the reimplementation of this system is non-trivial and there is no software available for the task, such a direct assessment of this effect is not available. Nonetheless, given the simplicity of our IR system, we suspect that our method of compounding performs significantly better. Systems in 2006 are usually much better, but this is primarily due to improvements in ranking and retrieval strategies. These improvements might in principle neutralize or make irrelevant any differences in compounding, however, the fact that most systems use decompounding allows us the conjecture that our advantage would carry over to more sophisticated IR systems.

We speculate that if some problems with guessing are mended, it might turn out to be more beneficial even when the full lexicon is used. As it stands guessing has only a non-significant positive effect.

Table 4 shows retrieval performance when POS tagger is used. We can see the unexpected result that contextual disambiguation has a negative effect on performance. Retrospecively, it is not too surprising that POS tagging brings no

Table 4. Result of different stemming methods *after* POS-tagging

	year	**MAP**	**P10**	**ret/rel**
without decompounding	2005	0.3289	0.3300	814/939
	2006	0.2855	0.3880	1009/1308
with decompounding	2005	0.3861	0.3740	893/939
	2006	0.3416	0.4360	1120/1308

improvements. First, for most of the tokens, there is nothing to disambiguate, either the analysis itself is unique or the lemmas belonging to the alternatives are identical. An example for the latter is so called syncretisms, situations where two different paradigmatic forms of a stem (class of stems) coincide. This is exemplified by the 1st person past indicative forms which are ambiguous between definite and indefinite conjugation.[4] So the effort of the tagger to make a choice based on context is futile for retrieval. In fact, some of the hardest problems for POS tagging involve disambiguation of very frequent homonyms, such as *egy* ('one/`NUM`' or 'a/`DET`'), but most of these are discarded by stopword filtering anyway.

Using the POS tagger can even lead to new kinds of errors. For example, in Topic 367 of 2005, the analyzer gave two analyses for the word *drogok*: `drog/PLUR` inflected and the compound *drog+ok* ('drug-cause'); the POS tagger incorrectly chooses the latter. Such errors, though, could be fixed by blocking these arguably overgenerated compound analyses.

4 Conclusion

The experiments on which we report in this paper confirm that a lemmatization in Hungarian greatly improves retrieval accuracy. Our system outperformed all CLEF 2005 systems that use algorithmic stemmers despite its simplicity. Good results are due to the high-coverage lexical resources allowing decompounding (or lack thereof through blocking) and recognition of various derivational and inflectional patterns allowing for agressive stemming.

We compared two different morphological analyzer-based lemmatization methods. We found that contextual disambiguation by a POS-tagger does not improve on simple local heuristics.

Our Hungarian lemmatizer (together with its morphological analyzer and a Hungarian descriptive grammar) is released under a permissive LGPL-style license, and can be freely downloaded from `http://mokk.bme.hu/resources/ir`. We hope that members of the CLEF community will incorporate these into their IR systems, closing the gap in effectivity between IR systems for Hungarian and for major European languages.

References

[1] Halácsy, P.: Benefits of deep NLP-based lemmatization for information retrieval. In: Working Notes for the CLEF 2006 Workshop (September 2006)

[2] McNamee, P.: Exploring New Languages with HAIRCUT at CLEF 2005. In: Peters, C., Gey, F.C., Gonzalo, J., Müller, H., Jones, G.J.F., Kluck, M., Magnini, B., de Rijke, M., Giampiccolo, D. (eds.) CLEF 2005. LNCS, vol. 4022, pp. 21–23. Springer, Heidelberg (2006)

[4] Finite verbs in Hungarian agree with the object in definiteness. If the verb is intransitive or the object is an indefinite noun phrase, the indefinite conjugation has to be used.

[3] Savoy, J., Berger, P.Y.: Report on CLEF-2005 Evaluation Campaign: Monolingual, Bilingual, and GIRT Information Retrieval. In: Peters, C., Gey, F.C., Gonzalo, J., Müller, H., Jones, G.J.F., Kluck, M., Magnini, B., de Rijke, M., Giampiccolo, D. (eds.) CLEF 2005. LNCS, vol. 4022, pp. 21–23. Springer, Heidelberg (2006)
[4] Goñi Menoyo, J.M., González, J.C., Vilena-Román, J.: Miracle's 2005 approach to monolingual information retrieval. In: Peters, C., Gey, F.C., Gonzalo, J., Müller, H., Jones, G.J.F., Kluck, M., Magnini, B., de Rijke, M., Giampiccolo, D. (eds.) CLEF 2005. LNCS, vol. 4022, Springer, Heidelberg (2006)
[5] Tomlinson, S.: Europan Ad hoc retireval experiments with Hummingbird TM at CLEF 2005. In: Peters, C., Gey, F.C., Gonzalo, J., Müller, H., Jones, G.J.F., Kluck, M., Magnini, B., de Rijke, M., Giampiccolo, D. (eds.) CLEF 2005. LNCS, vol. 4022, Springer, Heidelberg (2006)
[6] Savoy, J.: Report on CLEF-2003 monolingual tracks: Fusion of probabilistic models for effective monolingual retrieval. In: Peters, C., Gonzalo, J., Braschler, M., Kluck, M. (eds.) CLEF 2003. LNCS, vol. 3237, Springer, Heidelberg (2004)
[7] Tordai, A., de Rijke, M.: Hungarian Monolingual Retrieval at CLEF 2005. In: Peters, C., Gey, F.C., Gonzalo, J., Müller, H., Jones, G.J.F., Kluck, M., Magnini, B., de Rijke, M., Giampiccolo, D. (eds.) CLEF 2005. LNCS, vol. 4022, Springer, Heidelberg (2006)
[8] Trón, V., Gyepesi, G., Halácsy, P., Kornai, A., Németh, L., Varga, D.: Hunmorph: open source word analysis. In: Proceedings of the ACL 2005 Workshop on Software (2005)
[9] Trón, V., Halácsy, P., Rebrus, P., Rung, A., Vajda, P., Simon, E.: Morphdb.hu: Hungarian lexical database and morphological grammar. In: Proceedings of LREC 2006, pp. 1670–1673 (2006)
[10] Kornai, A., Halácsy, P., Nagy, V., Oravecz, C., Trón, V., Varga, D.: Web-based frequency dictionaries for medium density languages. In: Proceedings of the EACL 2006 Workshop on Web as a Corpus (2006)
[11] Halácsy, P., Kornai, A., Oravecz, C., Trón, V., Varga, D.: Using a morphological analyzer in high precision POS tagging of Hungarian. In: Proceedings of LREC 2006, pp. 2245–2248 (2006)
[12] Halácsy, P., Kornai, A., Németh, L., Rung, A., Szakadát, I., Trón, V.: Creating open language resources for Hungarian. In: Proceedings of Language Resources and Evaluation Conference (LREC04), European Language Resources Association (2004)