Parallel creation of gigaword corpora for medium density languages – an interim report

Péter Halácsy, András Kornai, Péter Németh, Dániel Varga

Budapest University of Technology Media Research Center
H-1111 Budapest Stoczek u 2
{hp,kornai,pnemeth,daniel}@mokk.bme.hu

Abstract

For increased speed in developing gigaword language resources for medium resource density languages we integrated several FOSS tools in the HUN* toolkit. While the speed and efficiency of the resulting pipeline has surpassed our expectations, our experience in developing LDC-style resource packages for Uzbek and Kurdish makes clear that neither the data collection nor the subsequent processing stages can be fully automated.

1. Introduction

So far only a select few languages have been fully integrated in the bloodstream of modern communications, commerce, research, and education. Most research, development, and even the language- and area-specific professional societies in computational linguistics (CL), natural language processing (NLP), and information retrieval (IR), target English, the FIGS languages (French, Italian, German, Spanish), the CJK languages (Chinese, Japanese, Korean), official languages of former colonial empires (Dutch, Portuguese, Russian), and a few dozen other *major languages.*[1] As is well known (Grimes and Grimes, 2000) the majority (56%) of the world's people speak neither one of the really large languages (100m speakers or more, accounting for about 40% of the world population) nor one of the 5,000 or so really small languages (500k speakers or less, accounting for about 4%) but rather some medium resource density language (MRDL).

The Less Commonly Taught Languages (LCTL) project of the Linguistic Data Consortium (LDC), see `http://projects.ldc.upenn.edu/LCTL`, aims at creating and sharing resources to support additional basic research and initial technology development in those languages that lack the high resource density of the major languages but have a million or more speakers. By this definition, there are only half as many LCTLs as MRDLs (roughly 240 out of 500), but the number of speakers covered is diminished only by about 4%, so the majority (according to statistics based on the SIL data, some 52%) speak an LCTL. The picture is greatly complicated by bi- and multilingualism, but overall there can be little doubt that MRD/LCT languages remain a central area for language resource building.

Since the central resource for modern CL/NLP/IR research is machine readable text, it is instructive to look at the list of wikipedias (`http://meta.wikimedia.org/wiki/List_of_Wikipedias`, February 2008) and live pages (`http://technorati.com/weblog/2007/04/328.html`, April 2007) as a proxy for widely available material. While the picture is somewhat distorted by artificial languages (Esperanto, Volapük, Basic English) which have enthusiastic wikipedia communities but relatively few webpages elsewhere, and by regional languages in the European Union (Catalan, Galician, Basque, Luxembourgish, Breton, Welsh, Lombard, Piedmontese, Neapolitan, Sicilian, Low Saxon, Occitan, Asturian, etc), which enjoy considerable language policy support, the ranking based on contemporary wikipedias and live pages shows the same skew toward the major languages as Comrie's decades old ranking:

lg. group	*% speaker*	*% wp article*	*% live page*
English	4.7	23.1	36
FIGS	3.9	21.4	9
CJK	16.0	7.1	45
Other major	22.3	33.2	5
Artificial	0	2.5	0
EU minority	0.3	4.2	0
All other	52.9	8.5	5

One could further refine the table by treating separately those languages like Ukrainian which are so closely related to some major language (in this case, Russian) that transferring language resources is feasible – assigning these to the "Other major" group would decrease the number of remaining wikipedia articles by about 15%, leaving less than 3,000 wikipedia articles per LCTL on the average. This average hides a considerable variation within MRDLs, from the high end of literacy (e.g. Slovenian, approx. 2m speakers, over 61,000 articles) to the low (e.g. Oriya, approx. 31m speakers, 540 articles, April 2008).

In earlier publications (Halácsy et al., 2004; Varga et al., 2005; Kornai et al., 2006) we outlined our methodology for obtaining high quality large-scale monolingual and parallel corpora for MRDLs and building more sophisticated LRs on top of these using our HUN* toolkit. Here, in Sections 2-3 we report on recent developments of integrating the toolkit with other FOSS resources to speed up three phases of the language resource building task: crawling, language identification, and tokenization. In Section 4 we consider the difficulties encountered in apply-

[1] In addition to those listed above, Comrie (1990) devotes full sections to Danish, Norwegian, Swedish, Rumanian, Polish, Czech, Slovak, Serbo-Croat, Greek, Hindi-Urdu, Bengali, Persian, Pashto, Hungarian, Finnish, Turkish, Arabic, Hebrew, Hausa, Tamil, Thai, Vietnamese, Burmese, Malay, Tagalog, Yoruba, Swahili, as well as to Latin and Sanskrit.

ing the HUN* tools to two LCT/MRD languages, Uzbek and Kurdish. For the major languages considered (Slovenian, Czech, Polish, Croatian, and Hungarian) in this interim report we can only provide speed, but not size, measurements, as the system is producing data at a rate that has outstripped our installed disk capacity (3 terabytes at the time of this writing). Our goal is to have new RAIDs installed and make the the full corpora accessible over the web at `http://mokk.bme.hu/multi` in time for the meeting in May. But for Uzbek and Kurdish, the entire LCTL resource package is complete, available both from the LDC and from the above URL.

2. Integrating crawling and language identification

The LCTL project of the LDC defined the resource package to contain the *absolute minimum*, both in terms of language material and support software, required to begin building a MT system between English and the target language. In particular, the monolingual corpus is only a quarter million words, with another quarter million coming from the target language side of a parallel corpus[2] included in the package. Yet, as we shall see in Section 4, the gap is very clear between the digital haves and have nots: while for major languages a gigaword monolingual target is quite easy to reach based on web crawls alone, for MRDLs one needs to include all kinds of non-web resources to reach the quarter million word target for monolingual, let alone parallel text. The LDC package also contains a dictionary with a minimum of 10k stems, aiming at 90–95% coverage of the monolingual corpus and a descriptive grammar that outlines the phonology, morphology, and syntax of the language, exemplifies the major constructions, and defines the major parts of speech (POS) in a manner consistent with the dictionary. As for support software, the package includes sentence- and word-level segmenters (see Section 3); a morphological analyzer and a POS-tagger using the same POS/MOR codes; and low-level support for transliteration between various encodings, in particular for proper (person) names. Since most MT systems work in conjunction with named entity recognition (NER), the package also includes a NER system and (manually verified) NER/POS/MOR tagging of the monolingual text.

As there are only about half as many active top-level domains (TLDs) as there are MRDLs, it is no surprise that most MRDL material has to be found in TLDs that are only mildly predictive of the language. The situation of the major languages, where e.g. a crawl of `.fr` is guaranteed to yield a sufficient amount of French material, is atypical for MRDLs, where websites of the same language are often dispersed among several TLDs. To be sure, one could enhance a `.fr` crawl by looking at `.ca, .re`, and other TLDs, but this is hardly essential, while a Tatar corpus would be hard to produce based on any single TLD. Even where most material is in a single TLD, as would be for Oriya in `.in`, this needs to be separated from a large amount of other material within the same TLD. Since the easiest way to limit a crawl is by TLD, our first goal was to assess the impact the choice of crawler and language detection method will have on a language-targeted crawl.

Among the public domain crawlers, `heritrix` (see `http://crawler.archive.org`) is specifically designed with periodic TLD-sized crawls in mind, and Baroni and Kilgarriff (2006) have successfully used it to create high quality gigaword corpora of German and Italian. Yet in our experience, after a good initial period, `heritrix` is difficult to control and throughput declines significantly. This is not a big problem for major languages with highly predictive TLDs, since in the first few days of the crawl a sufficient amount of material can easily be collected[3], but for the general MRDL case the stability of the crawler is an important factor. After similar experiences with two other widely used crawlers, `nutch` and `larbin`, we settled on the WIRE crawler (Castillo and Baeza-Yates, 2005), which has high throughput and is very parallelizable. Results on our own crawler, `nut`, are reported in a companion paper, (Kornai and Halácsy, 2008).

Offline, language identification is best performed based on spellchecking the page (Halácsy et al., 2004), but at download time this method may be too slow. We experimented with two other methods, the frequent word coverage heuristic used in Baroni and Ueyama (2006), and character n-gram language identification. Clearly, the frequent word heuristic is a special case of the language identification by spellchecking method (with the spellcheck dictionary restricted to the most frequent few hundred words), and it does not have the same resolving power as the full spellcheck, especially for mixed language and low quality pages. One important issue in many MRDLs is the widespread use of *flat* text created on keyboards designed for a major language, with graphemes specific to the target language replaced by the most similar unaccented grapheme, *o* instead of *ö*, *õ* and *ő*, *c* instead of *č* and so on. Filtering based on the frequent (function)words will often let such pages through.

Both of these methods are fast enough to perform runtime checks: if the language criterion is not met the page is not saved and the links are not followed. In the `.hu` TLD this affects about one page in a thousand (discounting flash pages, error messages, charset problems and other issues easily detectable without language identification) but if we search `.ro` for Hungarian pages the proportion is radically different, less than one page in ten is kept. The key issue is to achieve reliable decision for the target language not just against major languages but also against close and easily confusable language or dialect variants that are not targeted. For Uzbek, this meant building n-gram models for the Cyrillic, Cyrillic-flat, and Latin characterset-encoding variants and of course for Russian. For Sorani Kurdish, this required not just a CP1256 and UTF8 n-gram models, but also models for Arabic (both CP1256 and UTF8),

[2] Composed of 175k words manually translated from the target LCTL to English, plus 75k words of English text that is kept fixed across LCTLs: 30k news, a 20k Elicitation Corpus (see Probst et al 2001), and 25k words of text other than news.

[3] The top ten languages account for about 84% of static pages (see `http://www.internetworldstats.com/stats7.htm`, November 2007) and for 95% of 'live' pages such as blogs and social networks.

Farsi (UTF8), Turkish (CP1254), and Kurmanji Kurdish (CP1252, CP1254, CP1256, ISO8859-9). We chose Gertjan van Noord's TEXTCAT n-gram language classifier (see www.let.rug.nl/~vannoord/TextCat) both because we found it to be best of breed, and because it has the same permissive license, GNU LGPL, as the HUN* tools.

While using the web as a corpus for driving CL/NLP/IR work (Resnik 1999) is attractive for major languages, finding sufficient material for MRDLs is still challenging. Even for major languages, automatic recognition of URL parallelism (Chen and Nie 2000) can have a surprisingly low yield, and to create a significant parallel corpus one needs to include resources such as literary text, religious texts, international laws, movie captioning, software internationalization files, bilingual magazines, and annual reports and webpages of multinational corporations. To be sure, most of this material eventually finds its way to the web, but to identify them, negotiate for copyright releases, download them, convert the format and normalize the character-set encoding requires manual labor with little potential for automation except for the last few steps. This 'harvesting' effort (see e.g. http://lodl.ldc.upenn.edu/MRDL/Tamil_harvest.html) has a major payoff, generally enabling the collection of two-three orders of magnitude more parallel material than could be found by automated means, and reflecting a much wider range of jargons, styles, and genres than pure web text (see Varga et al 2005). Manual harvesting, described in Section 4 for Uzbek and Kurdish, remains a necessary prerequisite to the more automated stages.

3. Tokenization

Most of the pages downloaded by the crawlers are html – for the moment we ignore PDF files, MS Word documents, and other common formats, although in the long run extracting the text from these would obviously enhance the corpus. Altogether, WIRE can download about 8-10 GB of raw html pages a day: in our experiments, we obtained 10.3 GB/day for Slovenian (.si), 8.3 GB/day for Czech (.cz), 7.5 GB/day for Polish (.pl), and 8.1 GB/day for Croatian (.hr). Not surprisingly, the Hungarian throughput was considerably better, 15.7 GB/day. The offline processing steps discussed in this section are two orders of magnitude faster, so the overall speed of the corpus collection process is effectively determined by crawl speed.

Removal of the html markup is performed by the hunnorm text normalizer. This processing step compresses the downloaded corpus by nearly a factor of four: the resulting text is 26.1% of the original for Slovenian, 25.9% for Czech, 28.9% for Polish, 30.1% for Croatian, and 29.6% for Hungarian. The next filtering step is characterset detection and normalization (currently to Latin-1 or Latin-2). At this stage, pages with mixed or unidentified encodings are discarded, but this does not decrease the data set significantly. The speed of normalization is over 400GB/day (input, producing 100GB/day output).

The huntoken tokenizer performs an extremely crude, language-independent sentence boundary detection step, which is easily fooled by abbreviations and date conventions. Still, the resulting chunks are useful for several corpus cleaning methods: we remove all chunks that do not end in standard sentence-final punctuation (period, exclamation point, or question mark), and we discard all pages that do not have at least four chunks remaining after this. The throughput of this step is over a 100GB/day, so pipelining hunnorm and huntoken makes good sense.

Next we compare pages chunk by chunk, and discard as duplicates all pages where at least half of the chunks appears on some other page. Unlike Baroni and Kilgarriff (2006), we retain one page from each near-duplicate set. For major languages deduplication is a significant data reduction step, keeping only 28.9% of the Slovenian pages that were kept in the preceding steps, 25.6% of Czech, 19.9% of Polish, 31.7% of Croatian, and 30% of Hungarian. In contrast, MRDL corpora contain relatively few duplicates. Since deduplication is a two-pass algorithm it does not fit well in a pipeline, but the throughput, well over 20GB/hour, makes this unnecessary. Taken together, the postprocessing steps take less than an hour for a full day's WIRE crawl.

For major languages, we can set all thresholds very aggressively, so that after the quality checks, duplicate and language detection, only about a tenth of the original pages remain: 11% for Slovenian, 7% for Czech, 5.8% for Polish, 11.6% for Croatian, and 17% for Hungarian. In less than a month we created five high-quality corpora a 100 m words each (the cleaned Hungarian corpus, over a billion words in over 100m chunks, is already surpassing the one we published in 2004), and these are eminently suitable for corpus work.

Another area where automated techniques work reasonably well is POS tagging. Once the tagset is developed (a task currently beyond the power of unsupervised clustering and thus requiring human intervention), the hunpos tagger can be trained to a level of accuracy on 10k words that is more than sufficient for bootstrapping the process, see http://code.google.com/p/hunpos/wiki/RelatedPapers. Named entity recognition, using the hunner maxent tagger (Varga and Simon 2007), similarly proceeds from a small, entirely manually created training corpus to larger, bootstrapped NER-tagged corpora.

4. Uzbek and Kurdish

For the major languages it is generally trivial to find nation-states that use them as the dominant (official or unofficial) language. This implies not only the existence of at least one, and often several, TLDs that can be expected to contain dominantly material from that language, but also the existence of institutions ranging from national academies in charge of standardized orthography to newspapers of record (at this point typically publishing online editions as well), digital archives, legislatures and cabinet offices publishing minutes, national and international laws and regulations, and in general a vigorous online presence.

In this regard, Uzbek is among the top quartile of MRDLs, being the official language of Uzbekistan, where it is spoken by about three quarters of the 27m population. We restricted our work to the standard (Northern) dialect, though it should be noted that the Southern dialect is also spoken by over a million people, mostly in Uzbekistan and the neighboring Afghanistan. The standard orthography is Latin-based, but there is a significant amount of material either in Cyrillic-based Uzbek, which has characters outside the core Russian Cyrillic alphabet, and in 'flat' Cyrillic, which uses only the Russian characters. While we had no trouble finding three times the required minimum of monolingual Uzbek text, it is worth noting that the three major sources, the BBC Uzbek publications, the opposition journal Harakat, and Wikipedia, use three different character-encodings: Uzbek Cyrillic, flat Cyrillic, and Latin. We found it useful to develop transliteration into an extended Latin encoding that permits lossless conversion to/from Uzbek Cyrillic, because conversion from Uzbek Cyrillic to the official (Latin) orthography is lossy.

Also, even though a TLD `.uz` exists, and the Uzbek wikipedia has over 500 users (over 6k articles), a crawl of `.uz` yielded only 100k words of useful text, less than 1/6 of the material collected from the three major sources which are outside the `.uz` domain. Parallel material was practically nonexistent on the web, and had to be created based on the BBC and Harakat materials by a dedicated translation effort.[4] Fortunately, detailed descriptive grammars (in particular Bodrogligeti 2003) and a sizeable online dictionary (`http://uzbek.firespeaker.org`) already exist for the language.

Kurdish, by resource density, is in the bottom quartile of MRDLs. While it is clearly Indo-European (belonging in the Western Iranian group of the Indo-Iranian branch of the IE family), it is too far from any major IE language for resource transfer. Kurdistan, a geographic area comprised of land from Syria, Turkey, Iran and Iraq, is neither a unified political entity nor a TLD. The language situation is particularly difficult in Syria, where use of Kurdish is forbidden to the approx 1.5-2m speakers. In Turkey, the situation is better in that the laws and decrees banning the use of Kurdish have been withdrawn, but resentment of Kurdish by the Turkish-speaking majority persists. The two principal branches of modern literary Kurdish are Kurmanji, the language of the vast majority of Kurds in Turkey, Syria, Armenia, and Azerbaijan, the area designated by Kurdish nationalists as "North Kurdistan", with an estimated fifteen to seventeen million speakers, and Sorani, the language of most Kurds in Iraq (four to six million speakers) and Iran (five to six million speakers), the area designated as "South Kurdistan". Although the two are closely related, Kurmanji and Sorani are mutually intelligible only with difficulty in live contact, and not at all well in written form, because the two differ at the basic structural level as well as in vocabulary and morphology – for example, Kurmanji retains the case system that is no longer present in Sorani.

We concentrated on Sorani, which presents a harder problem from the resource perspective, not only because the number of speakers is smaller, but also because the area is farther from digital literacy. Sorani is generally written in Arabic-Persian script, with a few symbols added and some symbols reinterpreted, much like in Uzbek Cyrillic. The orthography is not entirely standardized, although there are important efforts in this direction, see e.g. `http://www.kurdishacademy.org`. There are several romanization schemes, including the official American Library Association/Library of Congress transliteration scheme, which is very detailed, but not very well suited for rapid text entry or data display as it relies heavily on diacritics. We settled on the kurdtUr scheme (see `http://www.cogsci.ed.ac.uk/~siamakr/Kurdish/kurdtur.html`), since it was designed from the ground up to work well in the context of partial internationalization. Since the predominant data entry, transmission, and display devices support only ASCII reliably, kurdtUr exploits the fact that there is now upper/lower case distinction in Arabic, and reuses some capital letters to encode some of the phonemes missing from the Latin alphabet. Conversion from kurdtUr to Arabic is unique but not lossless, since the Arabic script can omit the vowels (though Sorani written in Arabic script typically reuses the long vowel signs of the Arabic script to denote the (short) Sorani vowels). In the reverse direction, our software attempts to furnish the missing vowels. Unfortunately, there was no machine readable dictionary, and we had to key in printed wordlists to start the process, but pages harvested from `pukmedia.com` contributed well over 2m words. For the grammar, we relied heavily on Thackston's (2006) Sorani Reference Grammar, available at `www.fas.harvard.edu/~iranian/Sorani`. This grammar, just as Bodrogligeti (2003), is in many ways too detailed to be directly useful to the practicing computational linguist, who generally lacks the scholarly background required for the full appreciation of such works, and we found pedagogical grammars and exercises aimed at the language learner a great source of simpler examples.

5. Conclusions

Our experience with the machine-assisted process of assembling LR packages (data and tools deemed necessary for building MT systems) for MRDLs shows both that the HUN* toolkit can be rapidly adopted to new languages and that significant reliance on human effort is, for now, inevitable. In this regard, many of our assessments fly in the face of more enthusiastic research reports eager to declare victory.

First, automated collection of parallel corpora for MRD languages is not on the 5-10 year horizon, even though the method had been proposed nearly a decade ago (Resnik 1999), and has given rise to a whole cottage industry of web-based language resource building. Clearly, material must be collected from outside the TLD, even where there is a single TLD that offers good chance of success, and equally clearly, the material that can be detected based on URL parallelism remains a tiny fraction of the entire machine readable parallel text base.

Second, while model-based language identification techniques are reasonably mature, the current generation of clustering methods is too weak to actually supply these models with training material of sufficient breadth and quality. Language ID, including identification of the script

[4] It is perhaps worth noting that many of the translators were fearful of prosecution and refused working not only with the Harakat material that makes human rights a central concern, but also with the BBC reports that in Western society would be considered politically neutral.

and charset-encoding, remains a worthy challenge to unsupervised clustering work, especially if we expect the algorithm to be able to merge e.g. the Serbian (Cyrillic orthography) and the Croatian (Latin orthography) clusters of what is, after all, the same language. Similarly, automatic creation of POS tagsets remains a challenge, even though the first discovery procedures were proposed by Bloch, Nida, and Harris well over half a century ago.

Third, in spite of significant progress in this arena since the landmark Melamed (2001), no technique is quite ready to exploit parallel texts for high-quality dictionary building – the results are too noisy and the coverage is too weak for use in MT systems. While the human lexicographer can seemingly effortlessly undo the morphology, discard typos, foreign words, and other noise, and increase coverage by including translations for words that appear only once or twice in the whole corpus, automatic methods already have trouble with the core vocabulary (the top few thousand words by frequency).

Fourth, morphology remains a huge challenge, with the outermost (inflectional) layer of affixes slowly coming to the purview of machine learning algorithms, but the inner layers (derivation, compounding) remaining largely inaccessible. Crucially, the distinction between inflection and derivation is beyond the current power of automated systems, and has to be made by the grammarian on a case by case basis for each affix.

To some extent, these problems are interrelated: building better language identifiers would greatly aid the automated building of monolingual corpora, larger language-identified corpora would help us finding more parallel texts, better morphology would lead to better dictionary building, and so on. But for now, much of the critical information given in old-fashioned descriptive grammars, ranging from the transcription conventions to the morphotactics of the language, need to be supplied by humans.

The narrow goal of our paper was to describe how, in the face of such difficulties, human effort by computational linguists who do not themselves speak the target language, but have access to native speaker informants, can still be deployed efficiently to rapidly create language packages that are useful for higher-level natural language processing tasks such as syntactic parsing, information extraction, and machine translation. Clearly, progress can be made on each of the above areas without waiting for solutions from the others. But for the forseeable future, NLP remains the New York of Artificial Intelligence: if you can make it here you can make it anywhere.

Acknowledgments

Other major contributors to HUN* include Viktor Trón (`hunmorph`), László Németh and Bálint Gábor (`hunspell, huntoken`), Eszter Simon (`hunner`), and Csaba Oravecz (`hunpos`). The lion's share of the Uzbek and Kurdish work was performed by Viktor Nagy and András Rung, under a subaward from the Linguistic Data Consortium in the Less Commonly Taught Languages program – we thank Chris Cieri and Chris Walker for their constant help end encouragement.

6. References

Marco Baroni and Adam Kilgarriff. 2006. Large linguistically-processed Web corpora for multiple languages. In *Companion Volume to Proceedings of the European Association of Computational Linguistics*, pages 87–90, Trento.

Marco Baroni and Motoko Ueyama. 2006. Building general- and special-purpose corpora by web crawling. In *Proceedings of the 13th NIJL International Symposium, Language Corpora: Their Compilation and Application*, pages 31–40.

András Bodrogligeti. 2003. *An academic reference grammar of modern literary Uzbek.* LINCOM Europa, Munich.

Carlos Castillo and Ricardo Baeza-Yates. 2005. Wire: an open-source web information retrieval environment. In *Workshop on Open Source Web Information Retrieval (OSWIR).*

Jiang Chen and Jian-Yun Nie. 2000. Web parallel text mining for chinese-english cross-language information retrieval. In *NAACL-ANLP, Seattle*, pages 21–28.

Bernard Comrie, editor. 1990. *The World's Major Languages.* Oxford University Press.

Barbara F. Grimes and Joseph E. Grimes. 2000. *Ethnologue: Languages of the World.* SIL, fourteenth edition.

Péter Halácsy, András Kornai, László Németh, András Rung, István Szakadát, and Viktor Trón. 2004. Creating open language resources for Hungarian. In *Proceedings of Language Resources and Evaluation Conference (LREC04).* European Language Resources Association.

András Kornai and Péter Halácsy. 2008. Google for the linguist on a budget. In *LREC 2008 Web as Corpus Workshop proceedings*, page to appear.

András Kornai, Péter Halácsy, Viktor Nagy, Csaba Oravecz, Viktor Trón, and Dániel Varga. 2006. Web-based frequency dictionaries for medium density languages. In *Proceedings of the EACL 2006 Workshop on Web as a Corpus.*

I. Dan Melamed. 2001. *Empirical Methods for Exploiting Parallel Texts.* MIT Press.

Katharina Probst, Ralf Brown, Jaime Carbonell, Alon Lavie, Lori Levin, and Erik Peterson. 2001. Design and implementation of controlled elicitation for machine translation of low-density languages. In *Proc. MT 2010 Workshop, 8th MT Summit*, Santiago de Compostela, Spain.

Philip Resnik. 1999. Mining the web for bilingual text. In *Proc. 37th ACL*, pages 527–534, University of Maryland.

Dániel Varga and Eszter Simon. 2007. Hungarian named entity recognition with a maximum entropy approach. *Acta Cybernetica*, 18(2):293–301.

Dániel Varga, László Németh, Péter Halácsy, András Kornai, Viktor Trón, and Viktor Nagy. 2005. Parallel corpora for medium density languages. In *Proceedings of the Recent Advances in Natural Language Processing 2005 Conference*, pages 590–596, Borovets. Bulgaria.