Learning Language Identification Models: a Comparative Analysis of the Distinctive Features of Names and Common Words

Stasinos Konstantopoulos

NCSR 'Demokritos'
Athens, Greece
konstant@iit.demokritos.gr

Abstract

The intuition and basic hypothesis that this paper explores is that names are more characteristic of their language than common words are, and that a single name can have enough clues to confidently identify its language where random text of the same length wouldn't. To test this hypothesis, n-gramm modelling is used to learn language models which identify the language of isolated names and equally short document fragments. As the empirical results corroborate the prior intuition, an explanation is sought for the higher accuracy at which the language of names can be identified. The results of the application of these models, as well as the models themselves, are quantitatively and qualitatively analysed and a hypothesis is formed about the explanation of this difference. The conclusions derived are both technologically useful in information extraction or text-to-speech tasks, and theoretically interesting as a tool for improving our understanding of the morphology and phonology of the languages involved in the experiments.

1. Introduction

Language identification is performed on different levels, from the acoustic and prosodic to the phonotactic or graphotactic, and has found various application in speech synthesis, information extraction and data mining.

Leaving aside language identification at the acoustic and prosodic level, we shall concentrate on identifying the language of a string of phonemes or graphemes. In fact, all of the methods and experiments presented here operate on graphemes, but this choice is driven by data availability rather than any underlying assumption that cannot be circumvented.

We further concentrate on identifying the language of a single name, even when it is in isolation or in a document written in a different language. This is particularly interesting for language technology applications such as named-entity recognition or automatic transliteration, especially when spotting names transliterated into different orthography systems, e.g. spotting English-language named-entities in Chinese newspapers.

The intuition and basic hypothesis that the work presented here tests, is that names are more 'characteristic' of their language than common words are, and that a single name might have enough clues to confidently identify its language, where a common word of the same length wouldn't.

The paper is structured as follows: first an overview of the literature in language identification is provided, both in the framework of text categorization and for identifying the language of a single named entity in isolation (Sect. 2.). Then, in Sect. 3. we present an experimental setup for comparing name and generic-text language identification, the results of which are analysed in detail in Sect. 4. and 5.. Finally, conclusions and future research directions are discussed in Sect. 6..

2. Background

Guessing the language of a document falls under the larger area of *text categorization*, which aims at classifying a document as belonging to one (or more) out of certain, predefined categories or subject codes. Document language is one of the possible dimensions of categorization, interesting for various document organization, data mining, and information extraction tasks.

2.1. Document-level categorization

In their seminal paper, Cavnar and Trenkle (1994) report experiments on language categorization using a simple n-gram frequency algorithm. The language models consist of frequency counts of n-grams (up to 5-grams) for various languages. To classify a document, the frequency counts of n-grams in the document are calculated and their distribution compared against the distribution of n-grams in the language models. The model with the smallest distance from the distribution of the document, is assumed to be the language of the document.

This algorithm was tested on Usenet postings from the `soc.culture` newsgroup hierarchy. An eight-language corpus was generated semi-automatically: a first pass operated under the assumption that the postings are in the language of the country or region under discussion in each newsgroup, and at a second pass discrepancies between the newsgroup's default language and the system's prediction were manually resolved.

With the 400 most frequent n-grams retained in the models, and postings of at least 300 bytes of length, the system classified the test set almost perfectly, achieving an accuracy of 99.8%. The authors also report an accuracy of 99.3% for postings that are under 300 bytes, without providing any further details of how accuracy drops with shorter test documents.

Cavnar and Trenkle's algorithm has seen various implementations and applications, the most notable probably being the TEXTCAT[1] implementation used in the SPAMASSASSIN[2] spam filter. The TEXTCAT distribution includes

[1] `http://www.let.rug.nl/~vannoord/TextCat/`
[2] `http://spamassassin.apache.org/`

language models for 69 languages and about 9 kbytes of training text in each language.

2.2. Word-level categorization

Language identification is very accurate even for texts as small as two or three hundred characters, but even so that is a long way from identifying the language of origin of single name, when seen in isolation.

Efforts at language identification for proper names originate in speech synthesis (Spiegel, 1985; Vitale, 1991; Font Llitjós and Black, 2001), with language identification used to adjust grapheme-to-phoneme rules. The typical approach is to improve an English-language speech synthesiser by training n-gram classifiers and using different pronunciation models for foreign names, depending on each name's origin.

Font Llitjós and Black (2001), in particular, note that language identification of isolated names is a difficult task, as they tried to manually tag 516 names and found that they could confidently tag only 43% of the data. For their speech synthesis experiment they used a simplification of the Cavnar and Trenkle algorithm which only counted 3-grams. They trained language models on general text (ranging from 255 thousand to 11 million words), and provided the classification results as features for the grapheme-to-phoneme models. Unfortunately they do not report results for the language identification part of their experiments.

Another field of application of the same general methodology is automatic transliteration of named-entities for the purposes of machine translation (Huang, 2005), except that here language identification adjusts transliteration models instead of grapheme-to-phoneme ones. In Huang's experiment languages were grouped together in clusters, guided by the effect each clustering had on the accuracy of the overall transliteration. The resulting clusters roughly corresponded to familiar language groupings (Chinese, Romance, English-and-Dutch, Nordic). Employing language identification models is reported to improve the accuracy of the overall task, but no results are provided for the language identification sub-task per se.

Finally, language identification is also pertinent to information-extraction tasks such as named-entity recognition. In this context it is important to be able to identify the original language of a named entity in order to be able to recognize transliterated named entities. Virga and Khudanpur (2003) report identifying references to English-language named entities in Chinese text. Their approach is to train a tri-gram model on Chinese transliterations of English names and use it to pick out English-language named-entities. Knowing that a string is an English word, the original orthography can be more accurately guessed.

3. Language Guessing Experiments

The data and experimental setup is the same as previously reported (Konstantopoulos, 2007), based on European names extracted by harvesting websites listing football players and their nationality. Mixed-language nationalities (e.g. Belgian and Swiss) where discarded and certain nationalities were combined under a single language (U.K. & Ireland and Czech Republic and Slovakia.[3]) Table 1 shows the resulting list of languages, number of names, and average name length. This dataset was complemented by the JRC-Acquis Multilingual Parallel Corpus (Steinberger et al., 2006), used to establish a basis for comparison between names and common words.

Language	Names	Avg. Len.
German	2608	7.8
English	1132	7.5
French	1067	7.7
Italian	1042	8.2
Polish	944	8.6
Spanish	824	7.4
Dutch	746	7.5
Czech-Slovak	579	7.2
Swedish	542	8.6
Danish	501	8.2
Portuguese	418	6.3

Table 1: Corpus size statistics for surnames.

It should also be noted that the original corpus provides full names without any indication of how they should be split into their first name/last name components, so, for names with more than two parts, the last part was assumed to be the last name. This assumption makes the task slightly more difficult, since it removes language-specific surname prefixes like *van* and *della*, but is accurate in most cases since middle names are far more widespread than surname prefixes or double surnames.

Furthermore, all diacritics used in Latin-based scripts were dropped, since some are sufficient to considerably narrow the problem down or even identify a single language (e.g. Czech *ř*). This creates a performance mis-balance in favour of orthographies that prefer grapheme clusters instead of overloaded characters, as, for example, tell-tale clusters such and German *sch* or Polish *rsz* are retained, but characteristic graphemes such as *ř* or *š* in Czechoslovak are simplified to *r*. This 'injustice,' however, doesn't influence the result presented below, as we are interested in comparing general-language models versus name-specific models; as long as the handicap in any language is dealt in both models, the comparison results remain valid.

As a first setp, common words models trained on JRC data were used to predict the language of very short strings, comparable in length with a single surname (Table 1), as well as of actual names.

At a second step, the full name and last name datasets were used to train and test name models, using 10-fold cross-validation. N-fold cross-validation is a methodology for evaluating a hypothesis when there is not enough data to obtain both a training and a test set, but the same data has to be used for both training and validation, while at the same time guaranteeing the independence of the training and the validation process. The original set is partitioned into N

[3]Preliminary experiments have shown Czeck and Slovak *names* to be practically indistinguishable, despite the substantial differences between the Czeck and Slovak *languages*.

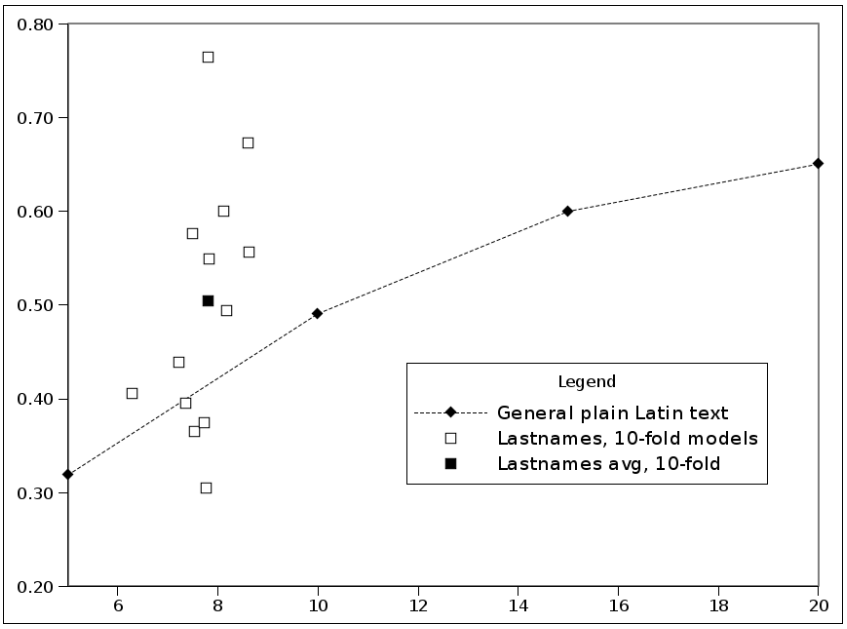

Figure 1: Graph plotting F-score of language identification against string length. The two lines plot language identification performance over general text of fixed length. The outlined square marks show F-scores per language, against average name length of the language. The filled square mark averages these last results over all languages.

subsets, of which one is retained as testing data and the remaining $N-1$ are used as training data. Training and testing is repeated N times (the folds), with each of the N subsets used exactly once as testing data. The N results from the folds are averaged to produce a single estimation.
Comparing the results of the JRC models and the names models, tested over names, we see that training models specific to names has a most profound effect on performance, as shown in Figure 1. This graph shows the, so to speak, relative 'discriminative density', of names and common words: last names are shown to carry a lot more potential per character than common words, as their average length is just under 8 characters, but can be predicted as accurately as common words of about 11 characters.

4. Derivational Morphology Analysis

Besides the purely methodological, and expected, result that one can get better performance by training name models on names, a more interesting theoretical question is the *reasons* why this is the case. One can think of various possible reasons, with varying degrees of theoretical interest. One of the most mundane analyses, for example, would be that names, being a more closed set that common words, offer themselves to over-specific modelling where high accuracy is achieved at the expense of generalization. Fortunately, such a hypothesis can be immediately refuted by the data in Table 2, as language models trained on names are, if anything, using shorter n-grams than the ones trained on common words.
This allows us to turn our attention to more interesting hypotheses, involving morpho-phonological or graphotactic features that (at least) European names possess, making language identification easier. Intuition suggests that surname forming typically involves a *son of* derivation from first names, so that different languages can be easily recognized by such suffixes. Comparing, for example, Thomas*sen*, Thom*son*, Tomas*evic*, and Tomas*evicz* one can immediately guess a Swede, an Englishman, a Czech, and a Pole.
This intuition was tested by counting what fraction of the n-grams in the model includes the end-of-word symbol and thus only matches word suffixes. Comparing the two sides of Table 3 one can easily see that suffixes are not more important for name models than they are for common word models. In fact, they are a slightly smaller fraction of the overall number of n-grams than in generic models.
A refinement of this hypothesis is that a smaller number of distinct suffixes might have a wider distribution in names. That is, that the derivational morphology creating surnames involves fewer suffixes applied to larger number of name instances so that, even though they are fewer, the relevant n-grams would have a heavier contribution to the language identification scores. This hypothesis was tested by adding the weights with which suffix n-grams contribute to a language's score, reflecting the number of *instances* of these n-grams in the training data. Again, as shown in Table 4 suffixes are not more important for name models than they are for generic word models.
An interesting observation can be made by comparing Tables 3 and 4: suffixes are 'heavier' n-grams, as smaller fractions of distinct n-grams cover considerably larger fractions o n-gram instances. This shows that suffixes are important discriminators, which is to be expected as in all languages involved morphological markers are mostly suffixes.[4] However, this is true for both names and generic words, and so cannot be the explanation for the higher ac-

[4]The notable exceptions being verbal inflectional morphology of German and Dutch. Indeed, *ge-* appears prominently in the

	Common word models						Surname models					
Lang.	n-gram length					Avg	n-gram length					Avg
Model	1	2	3	4	5	Len	1	2	3	4	5	Len
Czchslvk	40	187	108	42	23	2.55	25	192	141	34	8	2.52
Polish	43	187	108	42	20	2.52	24	174	135	43	24	2.67
German	48	164	118	52	18	2.57	25	212	120	37	6	2.47
Danish	43	160	124	51	22	2.62	26	177	124	46	27	2.68
Swedish	45	142	107	66	40	2.79	27	185	116	45	27	2.65
Dutch	37	170	121	52	20	2.62	25	195	143	31	6	2.50
English	36	184	129	39	12	2.52	26	212	142	19	1	2.39
French	43	162	123	52	20	2.61	29	203	152	15	1	2.39
Portug	45	159	122	54	20	2.61	26	167	164	36	7	2.58
Spanish	44	156	124	55	21	2.63	26	180	160	31	3	2.51
Italian	42	161	124	49	24	2.63	22	159	168	44	7	2.64
SUM	466	1832	1308	554	240	2.61	281	2056	1565	381	117	2.54

Table 2: Distribution of n-gram lengths in common word and surname models. The table shows the number of distinct n-grams in the model for each value of n, regardless of the number of instances of each n-gram found in the training data.

	Common word models						Surname models					
	1	2	3	4	5	All	1	2	3	4	5	All
CS	0.02	0.11	0.31	0.35	0.29	0.16	0.02	0.08	0.22	0.62	0.80	0.17
PL	0.02	0.10	0.28	0.41	0.50	0.16	0.02	0.07	0.16	0.28	0.41	0.13
SE	0.03	0.10	0.25	0.47	0.56	0.19	0.02	0.09	0.13	0.29	0.32	0.12
DA	0.02	0.11	0.25	0.53	0.86	0.20	0.02	0.09	0.14	0.41	0.56	0.15
DE	0.02	0.09	0.21	0.46	0.57	0.17	0.02	0.09	0.19	0.63	0.80	0.15
NL	0.03	0.09	0.20	0.42	0.40	0.18	0.02	0.08	0.19	0.67	1.00	0.15
EN	0.03	0.10	0.32	0.47	0.50	0.21	0.02	0.07	0.26	0.72	1.00	0.15
FR	0.02	0.08	0.31	0.57	0.67	0.21	0.02	0.07	0.26	0.47	0.00	0.14
PT	0.02	0.07	0.30	0.57	0.70	0.21	0.03	0.06	0.26	0.46	0.62	0.16
ES	0.03	0.07	0.27	0.50	0.77	0.20	0.02	0.06	0.25	0.35	0.33	0.14
IT	0.03	0.08	0.28	0.52	0.64	0.21	0.03	0.03	0.18	0.55	1.00	0.15

Table 3: Fraction of distinct n-grams that include the end-of-string, i.e. match suffixes.

curacy of the name models.

5. Distinctive Features

Having introduced the notion of an n-gram's 'quality' or 'usefulness' as a language discriminator, we turn our attention to a way to quantify this notion so that models can be compared based on how good discriminators they comprise. This quantification is based on the notion of *information content* and *entropy*.

Entropy is a measure of the lack of order, originally introduced in thermodynamic systems. Shannon (1948) transferred the concept in information theory, defining it as the expected (on average) number of digits required to encode a message, using the most efficient encoding possible. For an alphabet of k distinct symbols, appearing with relative frequencies $p_i, i = 1..k$, the *information content* of symbol i is $-\log p_i$ bits. The *entropy* of the encoding is then the weighted average of the information contents of all symbols:

$$H = -\sum_{i=1}^{k} p_i \log p_i$$

and estimates the number of digits necessary to transmit each symbol. Low values of entropy imply a high level of organization and the existence of patterns in the signal.

Given the appearance of an n-gram, we consider the predicted language the 'signal' that needs to be transmitted. The information content of each language is, then, the negative logarithm of the probability with which this language is predicted given the appearance of the n-gram; this probability is estimated by the weight assigned to the n-gram in each language's model, since heavier weights for a language make it more likely that this language will be predicted. We can now calculate the entropy of the n-gram as a whole, and use this to estimate the overall 'transmission efficiency,' i.e., how good the n-gram is at restricting the choice of language symbols that might be transmitted, so that the signal can be compressed.

To make this clearer, compare the distribution of the n-grams *e* and *rz* (Figures 2 and 3). The former is commonly

generic language models of both languages, but is (also as expected) absent from the name models. The Gaelic derivational prefixes Mac- and Mc- appear in the English surnames, but not frequently enough to even appear in the n-grams of the English names model, with 83 and 31 instances in 1132 names, resp.

	Common word models						Surname models					
	1	2	3	4	5	All	1	2	3	4	5	All
CS	0.31	0.20	0.37	0.46	0.17	0.28	0.28	0.18	0.39	0.75	0.84	0.27
PL	0.28	0.18	0.32	0.53	0.56	0.25	0.23	0.15	0.28	0.50	0.50	0.23
DE	0.28	0.19	0.33	0.60	0.68	0.27	0.25	0.16	0.32	0.66	0.84	0.24
DA	0.29	0.19	0.37	0.63	0.88	0.28	0.25	0.16	0.28	0.64	0.62	0.25
SE	0.30	0.20	0.38	0.65	0.68	0.29	0.23	0.15	0.26	0.53	0.54	0.23
NL	0.31	0.20	0.34	0.59	0.50	0.30	0.26	0.17	0.31	0.73	1.00	0.25
EN	0.33	0.21	0.41	0.66	0.72	0.32	0.26	0.17	0.37	0.77	1.00	0.25
FR	0.32	0.21	0.45	0.73	0.67	0.31	0.26	0.16	0.32	0.50	0.00	0.23
PT	0.33	0.21	0.44	0.68	0.76	0.32	0.28	0.17	0.35	0.55	0.77	0.26
ES	0.33	0.21	0.44	0.71	0.80	0.33	0.27	0.17	0.33	0.39	0.38	0.25
IT	0.35	0.21	0.41	0.68	0.73	0.33	0.25	0.15	0.31	0.70	1.00	0.24

Table 4: Fraction of n-gram instances that include the end-of-string, i.e. match suffixes.

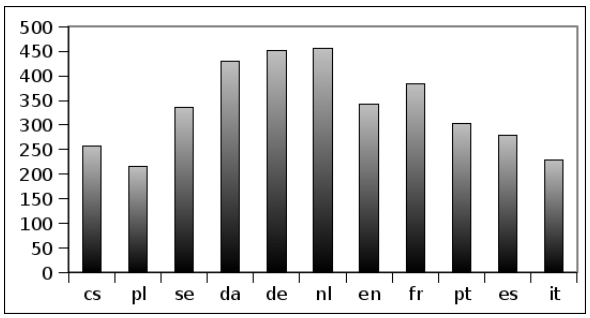

Figure 2: Distribution of occurrences of the unigram *e*. The numbers are normalized to represent the per-mil fraction of all words in *all* languages where it appears as a unigram of a *given* language. The entropy of this distribution is 1.79

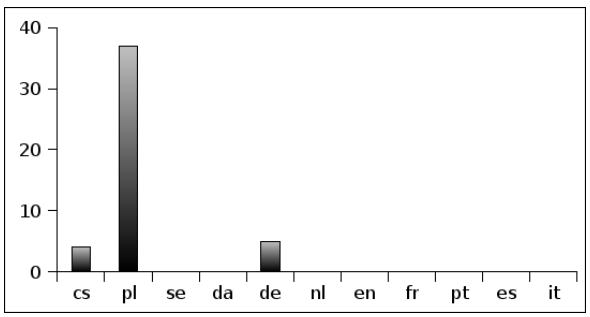

Figure 3: Distribution of occurrences of the bigram *rz*. The numbers are normalized to represent the per-mil fraction of all words in *all* languages where it appears as a unigram of a *given* language. The entropy of this distribution is 0.04

found across all languages, so that its presence does not help much compress the encoding of the language it predicted, as all languages are more or less likely to be predicted, and this unigram has a high entropy. The bigram *rz*, on the other hand, is very informative and drastically sharpens the distribution of possible predictions in its presence, so that it has a very low entropy.

After calculating and adding up the entropy of all n-grams in the generic language models we find a total of 934.3, versus 932.4 for all n-grams in the surnames models. Again, this difference is too small to account for the observed increase in prediction accuracy.

	Common	Names	
	Entropy	Entropy	F-score
PL	119.4	103.20	0.67
IT	152.8	141.28	0.60
EN	123.3	130.95	0.58
SE	126.2	135.78	0.56
DE	135.4	143.43	0.55
DA	147.2	137.84	0.49
CS	129.9	125.65	0.44
PT	150.7	141.12	0.40
ES	151.2	149.20	0.39
FR	155.6	144.39	0.37
NL	121.4	132.79	0.36
SUM	1512.9	1485.62	

Table 5: Entropies of distributions of n-grams of each language versus all the rest. The F-scores from Figure 1 are also shown.

A further refinement of this calculation is necessary, in order to take into account the fact that, in our example, the bigram in Figure 3 is good at discriminating Polish from the rest, but not particularly useful in discriminating, say, Spanish from Italian. In order to take this factor into account, we need to evaluate how informative an n-gram is for each language, as if 11 binary decisions were to be made, with each decision placing a different language in juxtaposition with the remaining 10.

We, thus, calculate for each n-gram in each language not the entropy of its distribution among all languages as we did before, but the sum of the entropies of these 11 one-versus-all distributions. Summing up these sums for each language, we get the results shown in Table 5.

What can be seen on this table is that, with the exception of Czeck-Slovak and Italian, the general tendency is for the entropy ranking to match the F-score ranking, a very strong indication that this explains the performance of the name models.

6. Conclusions and Future Research

The first conclusion drawn from the experiments presented in this paper is that people's names offer themselves for more accurate language identification than common words. This conclusion has been repeatedly hinted at in previous work on grapheme-to-phoneme conversion and transliteration, where a language identification pre-processing step resulted in dramatic performance increase on the main task. What is interesting to note is that the performance reported here is on a par with the performance of human annotators, who reported that they could only confidently predict a person's nationality in 43% of the data (cf. Section 2.2.).

What is, however, even more interesting and surprizing is that the expected and intuitive explanation that surname formation relies heavily on few and language-specific morphemes does not hold (Section 4.). The application of information theory concepts and techniques to quantifying featute quality, in pursuit of the features of names that make the difference, gives inconclusive but promising evidence: although the difference in the overall entropy of the models is not significant (934.3 vs. 932.4, cf. Section 5.) there seems to be a strong correlation between the predictive accuracy of the model and the one-versus-all quality measure given on Table 5.

This result can guide future research on the subject, as it can be used to identify the most promising features which to focus further experiments on. The correlation between the statistically extracted 'important' n-grams on the one hand, and n-grams that are known to be derivational morphemes on the other can also give interesting hints. The main effort in pursuing this direction would be the creation of the morphological resources, as surnames derivation is not a subject commonly treated computationally.

A second future research direction is that of using a cross-linguistically uniform and uninformed representation. In the experiments presented here a small step in this direction was taken by dropping all diacritics, so that there will be fewer chances for 'easy guesses' based on characters only found in a single language, but that creates the additional problem of accurately making all necessary the grapheme-to-phoneme conversions. An attractive alternative could be based on the assumption that transliteration to a completely different orthography to a large extend removes clues that are based on orthographic idiosyncrasies of the original language.

7. Acknowledgements

The idea of harvesting transfermarkt.de to extract people's nationalities was born four years ago, while discussing football with my good friend Evaggelos Litos.

The work described here was supported by the FP7-ICT project PRONTO.[5] PRONTO develops methodologies for the analysis and interpretation of textual, audio, and video data, aiming at the extraction of operational knowledge supporting and improving resource management.

8. References

William B. Cavnar and John M. Trenkle. 1994. N-gram-based text categorization. In *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, 11–13 April 1994*, pages 161–175.

Ariadna Font Llitjós and Alan W. Black. 2001. Knowledge of language origin improves pronunciation accuracy of proper names. In *Proc. of Eurospeech 2001, Aalborg, Denmark*.

Fei Huang. 2005. Cluster-specific named entity transliteration. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, British Columbia, Canada*, pages 435–442, Morristown, NJ, USA. Association for Computational Linguistics.

Stasinos Konstantopoulos. 2007. What's in a name? In Petya Osenova, Erhard Hinrichs, and John Nerbonne, editors, *Proceedings of Computational Phonology Workshop, International Conf. on Recent Advances in NLP, (RANLP), Borovets, Bulgaria, September 2007*.

Claude E. Shannon. 1948. The mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423, 623–656, July, October.

Murray F. Spiegel. 1985. Pronouncing surnames automatically. In *Proc. Conf. of the American Voice Input/Output Society*, pages 109–132.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proc. of the 5th International Conference on Language Resources and Evaluation (LREC'2006), Genoa, 24–26 May 2006*.

Paola Virga and Sanjeev Khudanpur. 2003. Transliteration of proper names in cross-lingual information retrieval. In Mari Broman Olsen, editor, *Proceedings of the ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition*, pages 57–64.

Tony Vitale. 1991. An algorithm for high accuracy name pronunciation by parametric speech synthesizer. *Computational Linguistics*, 17(3):257–276.

[5] http://www.ict-pronto.org/