Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 985–990
Marseille, 11–16 May 2020

Dialect Clustering with Character-Based Metrics: in search of the boundary of language and dialect

Yo Sato, Kevin Heffernan
Satoama Language Services, Kwansei Gakuin University
Kingston-upon-Thames, U.K., Sanda, Hyogo, Japan
satoama@gmail.com, kevin@kwansei.ac.jp

Abstract

We present in this work a universal, character-based method for representing sentences so that one can thereby calculate the distance between any two sentence pair. With a small alphabet, it can function as a proxy of phonemes, and as one of its main uses, we carry out *dialect clustering*: cluster a dialect/sub-language mixed corpus into sub-groups and see if they coincide with the conventional boundaries of dialects and sub-languages. By using data with multiple Japanese dialects and multiple Slavic languages, we report how well each group clusters, in a manner to partially respond to the question of what separates languages from dialects.

Keywords: clustering, dialects, similar languages, Japanese, Slavic languages, distance metric

1. Introduction

'A language is a dialect with an army and navy' is a well-known dictum attributed to sociologist Max Weinrich. It often happens that what are considered to be distinct languages appear more similar to each other than what are considered to be dialects. Our objective, however, despite what the sub-title might suggest, is *not* a pursuit of a clear distinction. Instead the idea in which this work is oriented is that such distinction should be dissolved into a gradient scale — i.e. that all groupings of languages or dialects are a matter of degree, relative to some metric.

In this exploratory work to we experiment with a set of metrics and feature representations on *shuffled* corpora — classified corpora jumbled into one— for clustering, to see how close the resulting clusters are to the original grouping. We use two sets of classified corpora, of 'dialects' (Japanese dialects) and of similar 'languages' (East/South Slavic languages), five-way classified respectively, and report how well —in relation to the conventional grouping— these dialects/languages cluster, in a manner in which to partially reply to the question of whether 'dialects' can be more different from each other than 'languages' are.

Clustering of languages/dialects is, in comparison to its supervised counterpart, their classification (or identification), a relatively less established field, presumably mainly due to its difficulty to achieve deployment-level performance. Given, however, the reality that digital language resources are often mixed in languages/dialects, the work on their clustering bears not just a theoretical but practical value, as the technique can be used for useful pre-processing to an NLP workflow.

Performance aside, another major theme of this work is data representation of a sentence. Clustering standardly requires a distance that can be computed on two given datapoints and they are represented typically by feature vectors, but what would be the best method for the purpose of dialect clustering? If it requires too much human effort to fill in the feature vectors, that would defeat the object of unsupervised learning. Furthermore, the representation for cross-lingual clustering like ours needs to be dialect/language-independent, as we cannot presuppose the linguistic/dialectal identity. We therefore employ character-level feature vectors, taking the Roman alphabet as a proxy character set for phoneme-level representation.

We will show that, with this relatively simple setting, we can achieve a reasonable set of results. In terms of the dialect / language question, a better set of results have in fact been achieved for Japanese dialects than, not just the Balkan languages but than between East and South Slavic languages. While this result by no means definitively shows that the first group is more internally similar than the second, given the exploratoty nature of our experiments, it can be considered a first step towards a universal metric upon which an objective grouping of languages / dialect may be achieved.

2. Related work

As stated, to the knowledge of the authors, unsupervised clustering for text processing is not an area that has been extensively studied. However, the use of characters for text processing is not a new idea and has been explored in the context of dialect processing, and some unsupervised techniques have started to emerge in dialect processing. Furthermore, unsupervised clustering *is* an actively pursued topic in the speech recognition community in a somewhat similar context to ours, as well as in biology, in a rather different context, i.e. the discovery of protein sequence patterns.

Dialect identification for text is clearly a closely related topic, which is studied actively for multi-dialectal language communities such as those of Arabic (Zaidan and Callison-Burch, 2013) and Chinese (Williams and Dagli, 2017). The varieties of English, which could also be considered variant dialects, have also been the target of investigation (Lui and Cook, 2013). These studies have generally invoked some form of character-level processing, be it embeddings or N-grams. Scherrer (2014) provides a 'pipeline' that invokes several methods including character-level MT, in a partly unsupervised approach. However, these works rely on the presence of the 'standard' form for which the dialects are variants, making them characterisable as transfer or adaptation approach, or semi-supervised modelling. In this broad sense, they are similar in spirit to ‘normalisation’ studies that have nothing to do with dialect, as in Han et al. (2011), in which the authors deal with ‘noisiness’ of social media content in an attempt to retrieve the ‘standard’ forms, or in Saito et al. (2014), where the authors try to group the orthographically normalisable variants.

In contrast, our study starts from scratch, and simply does not assume any ‘standard’ to which any particular group or sentences should be assimilated, or use any pre-trained model. It is the domain of speech where such pure clustering has drawn more interest, since the researchers take interest in clustering the individual realisation in articulation into groups, mainly those of accents. While accent identification could take the form of adaptation (the popular i-vector method, for example (Cardinal et al., 2015), there have been attempts to cluster from scratch, where the researchers use approaches such as Gaussian Mixture Models (Mannepalli et al., 2016), or in the recently more popular method of auto-encoder (Kamper et al., 2015). Such models however require a more continuous feature space. While it is conceivable to create a continuous feature space for texts, it would require some pre-processing to extract such features.

The rather unlikely domain from which we take inspiration most is bioinformatics. For a relatively discrete feature space like text, a very similar challenge is faced in *sequence clustering* that is used for discovery of the types of proteins from discrete sequences of amino acids. Amongst the possible options, we employ rather recent *Sequence Graph Transform* (SGT) (Ranjan et al., 2016), as it claims to be less length-sensitive than the popular alternatives such as UCLUST (Edgar, 2010).

Another area where a similar approach is taken is document clustering. For semantic, topic-based grouping, where unambiguous, one-to-one labelling is often difficult, the use of vectors is common to cluster documents instead of learning on a pre-labelled dataset (e.g. (Sharma and Dhir, 2009)). The difference of course is that their target is a document, and their preferred units are words. In a sense, our work can be characterised as doing what document clustering commonly does on the individual sentence level.

3. Experiment overview

The experiments are generally designed to cluster two shuffled corpora, each of which was originally classified into similar languages or dialects. As mentioned in the introduction, we use two groups of corpora, one Japanese, originally classified into five dialects, the other East/South Slavic, which has been originally classified into five languages. The implication is that the first represent a ‘supposedly dialectal’ group, while the second a ‘supposedly linguistic’ group, but we simply call them Japanese and East/South Slavic, to stay unbiassed.

Before embarking on the main experiment of intra-group clustering, we first report on the preliminary work to test the soundness of our approach, by checking the mean vector distances between groups, and test-cluster the whole datasets, i.e. Japanese and Slavic groups, to confirm it works *across* these groups (‘reference’ experiments).

We then move on to the main clustering experiments of trying and differentiating similar languages and dialects, where we use three method of distance computation, all character based, unigram, bigram and SGT, in the order of simplicity.

All these methods require the same feature dimensionality for datapoints. Thus it is incumbent on us to decide what character set, or *alphabet* as we will call it, is to be used. In their respective standard orthographies, Japanese and East/South Slavic languages employ very different alphabets indeed, and within the latter, there is a divide of Roman and Cyrillic characters. Given the fact that there is a one-to-one mapping system of Japanese and Cyrillic characters into the Roman ones, we have made the practical decision to use the latter. We then have the question of diacritics for the East/South Slavic group. In the main, we will use the de-accented ascii counterparts for all the diacritics, though we will show the results of using diacritics alongside. In addition, all characters have been lowercased before the experimentation. Therefore we mainly use a very restricted alphabet consisting of 28 characters, that is, 26 roman alphabet letters along with comma (‘,’) and full-stop (‘.’). Other punctuations are all normalised into one of them.

Importantly, we did not use the space character. That is, all the word boundaries have been deleted, making the sentence effectively a chain of printable characters. This is firstly to make use of characters as the proxy for phonemes in normal speech, which are not ‘separated’ with pauses, and secondly, to circumvent the difficulty of segmenting Japanese, which is not normally segmented, and for which there are segmentation ambiguities.

For clustering, we use three popular methods: KMeans, divisive (top-down) hierarchical clustering (HC), and agglomerative (bottom up) HC. There are some hyperparemeters to tune, which we will discuss in the experiment section below. Furthermore, there is a sparsity issue for sentences. That is, we cannot expect all the characters, or bigrams, to appear in a single sentence all the time. Therefore, we first impose the threshold of 100 characters on the sentence length. As this will not be sufficient, we will also use dimensionality reduction. We will discuss the details in the experiment section.

4. Data

4.1. Japanese dataset

The Japanese dialect-classified dataset comes from two sources. One is the Parallel Speech Corpora of Japanese Dialects (Yoshino et al., 2016) henceforth PCJD, consisting of four sets of sentences that each represent a dialect (Tohoku, Kansai, Chugoku and Kyushu). Each set consists in turn of 500 sentences, the translations by five native speakers of the dialects, of the Tokyo dialect equivalents. Thus PCJD, with the Tokyo dialect included, provides five pre-classified dialect corpora.

Since this set is not so large and is somewhat artificial, we supplemented it with a Twitter corpus the content of which we collected ourselves (Twitter Inc., present), more precisely a subset of it which has been identified and classified by humans into the five dialects as above. This counts another 300 each, and therefore altogether, we use the dialect-corpora in which each of the dialects counts 800, that is 4,000 altogether.

The corpora are in the standard Japanese orthography. As we use a Roman character set, they are first converted into roman characters using an automatic converter KAKASI (Kak, 2014). The Japanese punctuation characters are also converted into either a full-stop or comma, so that the characters fall within the 28-character alphabet we use. As mentioned earlier, we do not use space characters, and there is none in the Japanese orthography in the first place.

4.2. East/South Slavic dataset

The East/South Slavic dataset consists of the classified corpora used in the shared task of the ACL Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial) (Malmasi et al., 2016). The five languages represented are Czech, Slovak, Serbian, Croatian and Bosnian. The expectation, at least from the language family point of view, is that the first two and the last three are closer to each other respectively than between the groups. The datapoints used in the experiment count 1,000 each.

Here, apart from the removal of spaces, the main preprocessing measure concerns de-accenting the diacritics. As stated, this decision synchronises with the Japanese treatment stated above, which makes the inter-group comparison more meaningful. For the intra-group comparison, that is amongst the East/South Slavic languages, this treatment, abstracting the phonetic variations away, will make the clustering task a little harder. We will however show the results of the case of including diacritics.

5. Experiments

5.1. Vector representation of a sentence

As we discussed, as a universal method of representing a sentence, we use character-based vectors. Character composition of a sentence can be represented as a feature vector in a couple of ways. One possibility is an N-gram occurrence vector. The simplest will be a unigram vector, or frequency count of characters, which would constitute a so-called *bag of words* model, if a misnomer in our study. In this model, for the sequence *ACCB*, given a five-letter alphabet $\{A, B, C, D, E\}$, the vector will be $[1, 1, 2, 0, 0]$. This model will however only capture the presence of characters, not their sequential patterns. A bigram or longer N-gram will capture such patterns, but there are certain disadvantages here: N-gram modelling will make the Markov assumption (independence of series of N-grams) which may not be valid in our context, and will also have a combinatorial explosion when the N becomes large. A bigram vector will have, for a 26-letter alphabet, $26!/(26-2)!+26 = 676$ ($+26$ for the repetition of the same letter) in dimensionality, while a trigram one would have 15626. Furthermore, a large N would suffer from a sparsity problem: a certain sequence of characters may be either overrepresented or underrepresented in a small set of data.

Against the conventional method we compare the results obtained through SGT we quoted earlier, as a more sophisticated alternative. An SGT features a similar vector to that of bigram counts, with the identical dimensionality (and hence computationally very economical). That is, our features, or 'entries', would be bigrams, $AA, AB, AC, ...$ and so on. The key difference from the bigram count vector is that, as the name implies, it handles the *chain* of characters rather than its isolated occurrences, and the association of the two characters is expressed not by an integer but by a normalised real number. Here is the relevant equation from Ranjan et al. (2016), for a feature value Ψ for two characters, u and v, in a sentence s:

$$\Psi_{uv}(s) = \frac{\Sigma_{(u,v)\in\Lambda_{uv}(s)} e^{-\kappa|pos(u)-pos(v)|}}{|\Lambda_{uv}(s)|} \qquad (1)$$

where $\Lambda_{uv}(s)$ is the set of the pairs of u and v occurring in s and $pos(c)$ is the relative position of a character c in the sentence.

Notice that there is a hyperparameter κ, which controls the weights given to the positional distance between two chracters (the number of characters intervening them): the higher it is the less weight is given to a more positionally distant chain, i.e. two characters occurring farther apart. In a linguistic experiment like ours, we would want to have a relatively small weight to a distant chain. We will use the range of 10-20 over which we tune for the optimum.

We have used in this study all the three methods throughout, unigram, bigram and SGT, though we will mainly report the SGT results, as it turns out that it outperforms the other two consistently.

5.2. Dimensionality reduction

Another issue that concerns our representation of vectors is the dimensionality, which may be small for computation but is large enough to cause a sparsity problem *on the individual sentence level*. A single sentence is less likely than not to contain all the bigram patterns exhaustively of the language it belongs to. We address the issue to a limited extent by setting the threshold for a sentence length (at 100), but this is far from sufficient to ensure most of the character combinations will occur in a sentence. As they are, on average, about 25%, 34% and 28% of the feature entries are left at 0 respectively for our three metrics (unigram, bigram and SGT).

We therefore apply a common dimensionality reduction technique, Principal Component Analysis (PCA) to reduce the features and optimise a model, experimenting with a range of values empirically. The eventually chosen number is indicated by n below.

5.3. Mean distances between languages / dialects

By way of a preview to the potential viability of the proposed methods, we computed the mean statistics on the overall differences between languages/dialects before the main experiments to verify that there are differences to be discovered in the first place. First, the difference in the mean distances between the Japanese and Slavic groups is .218, which is clearly significant, at $p < .0005$ (t-test).

Table 1 shows the mean distances amongst the dialets/languages inside each of the groups in the form of a matrix, along with the significance levels (* for $<$ .05, ** for $<$ .01). It can be seen that the differences are mostly

	Tohoku	Kansai	Chugoku	Kyushu
Tokyo	**.103****	.095**	.039*	.043*
Tohoku	0	.075*	.035*	.044*
Kansai		0	.025*	.018*
Chugoku			0	.024*

	Slovak	Bosnian	Croatian	Serbian
Czech	.029* .031*	.094** .094**	.098** **.099****	**.099**** .089*
Slovak	0	.085* .089*	.072* .079*	.079* .087*
Bosnian		0	.019* .022*	.009 .014*
Croatian			0	.023 .028*

Table 1: Mean distances between languages / dialects

significant, reflecting the apparent distinctness for the human eye. While the differences may not be clear cut, this at least shows there are differences to be discovered: a necessary condition for the possible clustering success.

Here we mostly see the results that conform to the language family taxonomy: Czech and Slovak are more similar to each other than the Balkan languages are amongst themselves. For Japanese, there are some results against it, however. It is generally perceived that, in a manner that follows their geography, Kansai, Chugoku and Kyushu form a transitive 'chain' of similarity, that is, Kansai and Chugoku are similar, Chugoku and Kyushu are similar to a similar degree, but Kansai and Kyushu are not so similar. This common observation is *not* borne out. Instead, all the three pairs are almost as similar to each other in these results. We will later see these 'similarities' are carried over to the clustering results. On the other hand, the conventional observation that the dialect of Tohoku is dissimilar to any of the rest is borne out.

5.4. Reference case: differentiating Japanese and East/South Slavic

We start with the 'easy' case of separating Japanese and East/South Slavic to see the viability of our methods, the results of which are shown in the confusion matrix below.

	Japanese	Slavic	Recall
Japanese	3009/3880/3976	991/120/24	.753/.984/.994
Slavic	729/78/21	4271/4922/4979	.854/.984/.996
Precision	.804/.980/.995	.811/.976/.995	

We show all three results here, for the unigram, bigram and SGT encodings. In terms of the types of clustering algorithms (henceforth 'clusterers'), the results are with the Agglomerative HC with Ward linkage, which consistently performed 'best' over the other methods, i.e. KMeans and Divisive HC, though the margins were small. As can be seen, the clusters on SGTed vectors ($\kappa = 15$) achieved very good results with optimally-reduced features ($n = 30$), the bigram one not so much further behind ($n = 35$). The unigram method lags behind, and shows poor performance even for this clear-cut case. We might note here also that for the unigram model the dimensionality reduction does not lead to much improvement over the original 28. While we ran clustering on all the three encodings for the main experiment to be reported on below, since this diffrence margins are largely consistent, we will dispense with the unigram/bigram results and will show only the results with SGT from now on, though we will mention the differences occasionally observed on the different clusterers.

5.5. Main results: clustering similar languages and dialects

Table 2 shows the main results of clustering Japanese dialects and East/South Slavic languages respectively, in the form of confusion matrix. The figures in brackets in the Slavic group show the case of using diacritics.

We only show the 'best' results in terms of clusterers and hyperparameters to avoid clutter, but we might note here some general trends. First, the optimal kappa parameter and PCA counts, while they were not so different from the reference experiment in the Japanese group, were generally higher in the Slavic group (κ: $20 - 25$, PCA: $45 - 50$). Furthermore, different clusterers did produce more different results than the reference experiments, though generally speaking, the same clusterer, that is the Agglomerative with Ward linkage, produced the best results. KMeans generally produced comparable, though slightly worse, results. What was markedly different from the reference experiment is that there were occassionally great differences between different linkage methods, and occasionally the 'complete' linkage method outperformed. This is likely to be due to its robustness to outliers and propensity towards equally sized groups. This aspect of parameter tuning would warrant further investigation, though outside the scope of this work.

For Japanese, there is a clear 'winner', that is the Tohoku dialect, in a manner that conforms to the conventional observation. The Tokyo dialect fares well too, if not so well in precision. On the other hand the clusterer seems to struggle with differentiating the three Western dialects, Kansai, Chugoku and Kyushu. Nevertheless, as the scatterplot (Figure 1) shows, there are three-way cluster emerging, that is, Tohoku, Tokyo and the rest.

For the Slavic group, a similar picture is emerging, as two major groups (Czech and Slovak on one hand, the rest, Balkan, languages on the other) are better clustered, while the intra-group differentiation proves difficult. The Balkan clustering in particular seems hardly better than the chance level. In parallel we experimented on the data with diacritics left intact, and the absolute gain for the success cases is shown on the table. Keeping diacritics however did not help much, with only marginal improvements. There seems to be a dilemma here: dimensionality reduction means that finer grained features that could be manifested on the diacritics level is not highlighted, while without it, the sparsity

	Tokyo	Tohoku	Kansai	Chugoku	Kyushu	Recall
Tokyo	686	10	21	31	52	**.857**
Tohoku	20	670	33	28	49	.837
Kansai	29	40	496	112	123	.620
Chugoku	41	19	176	425	139	.531
Kyushu	89	72	101	180	358	.447
Precision	.793	**.826**	.599	.543	.496	

	Czech	Slovak	Bosnian	Croatian	Serbian	Recall
Czech	525(+23)	354	33	36	52	**.525**(+.26)
Slovak	431	459(+19)	33	28	49	.459(+.22)
Bosnian	29	40	376(+14)	234	321	.376(+.14)
Croatian	41	19	341	298(+20)	301	.298(+.00)
Serbian	89	72	277	180	285(+4)	.382(+.00)
Precision	.470 (+.03)	**.486** (+.02)	.354 (+.01)	.373 (+06)	.282 (+.00)	

Table 2: Clustering performance, languages / dialects

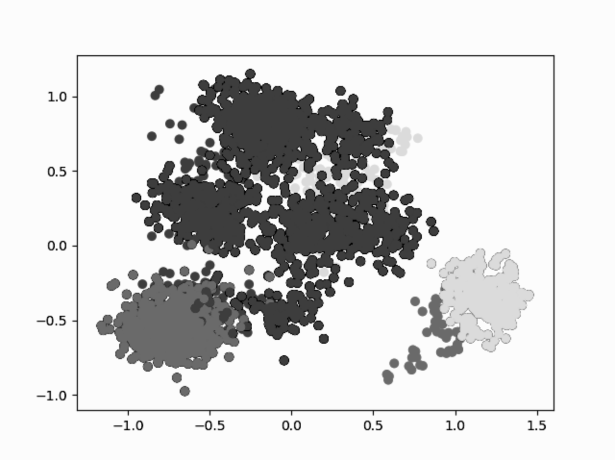

Figure 1: Cluster patterns for Japanese

takes over and lets the zero-valued features dominate.

In short, the clustering methods as experimented here work to a limited degree: limited, assuming that the distinction is mostly possible for humans, at least given diacritics for Slavic. 'Mostly' here may be the key to understanding the failure of clustering on finer points however. As we have found in Japanese, there are indeed cases that are not even clear to the human eye, and also, as we shall see shortly, there are features common to a subset of dialects. Such cases might cause the overall failure in clustering. In the next section we will test a measure to recover from such indistinguishable cases.

6. Error analysis, and a top-down enhancement for clustering

As we have seen, for close languages and dialects, they appear to resist the clustering generally possible to the humans with the methods employed here alone. The error analysis of Japanese sheds some light towards the possible remedies. The core problem is that on the level of individual sentences, any one sentence may be devoid of features that are distinct in that dialect or language. True, that the corpora are classified, which does imply that some distinctive feature is present in each sentence, but then, there is the possibility that more than one cluster may share that feature, if not with other clusters. In Japanese, the neighbouring dialects can have a significant amount of shared vocabulary, even where dialects generally differ. For example, the auxiliary *yoru* and *ja*, though not used in the standard Japanese, are both used frequently in the neighbouring Chugoku and Kyushu dialects.

	Tokyo	Tohoku	Kansai	Chugoku	Kyushu	Recall
Tokyo	690	10	21	29	50	.862(+.005)
Tohoku	19	671	33	28	49	.838(+.001)
Kansai	29	40	531	101	99	.663(+.043)
Chugoku	39	19	120	508	114	.635(+.103)
Kyushu	87	72	91	152	398	.497(+.050)
Precision	.798 (+005)	.826 (+.000)	.667 (+.067)	.619 (+.075)	.560 (+.064)	

Table 3: Improvements with SSR

The problem caused by this vocabulary sharing for the standard clustering is that, once such a pair is clustered together, there is no recovering from such 'mistakes': in fact no mistake is involved here on the level of the pair. What is required to make such recovery possible is a constraint applied to change the elements partially of the clusters as the process proceeds, as and when the wrongness of the pairing becomes apparent. We have therefore chosen, as such a constraint, to use the *sequence sharing rate*, or SSR. The intuition is that a dialect will have a *consistent* shared vocabulary, and hence, even if some words can happen to be shared across dialects, the substring sharing *as a whole* inside a dialect should be higher than across dialects. By shared sequence we mean a contiguous substring that is found in the target strings. We take the longest match. Therefore for example between *abcde* and *ijbcdk* it is *bcd*. We also take multiple matches if they exist but not repetitions in the same string, so for *abcdef* and *efabcef* we will have two shared sequences, *ef* and *abc*. Given a set of utterances U and a set of shared sequences that a set of shared sequences S, SSR is defined as follows:

$$SSR(U) = \sum_{s \in S} (len(s) \times 2)/|U| \qquad (2)$$

where $len(s)$ refers to the number of characters in shared sequence s. Notice we give more weights, proportionate to two, to longer shared sequences, given the likelihood that longer sequences contain words and phrases, which we are implicitly modelling.

Now, the clusterer suited to apply such a top-down constraint is a hierarchical one. For the divisive HC, in each iteration the split is made that makes the distance between the sub-clusters will be maximised. In this distance computation, instead of the simple cosine vector distance on datapoints, we interpolate the SSR, such that the average of SSR and vector distance will be maximised instead.

In Table 3 we report the resulting performances. The numbers in the bracket are the improvements over the model without this treatment. We have marked improvements in the Kansai/Chugoku/Kyushu clusters for Japanese, although not as much improvement was achieved for the Slavic group.

7. Final remarks and future tasks

We have shown that plausible clustering from scratch is possible for some conventional language / dialect groupings by means of character-based encodings. There are some difficult cases, like our Balkan languages, though we remain agnostic about whether this is due to the methods we employ, or whether there is no latent features to be discovered. We also showed that a certain metric may not point to what humans consider to be clear differences, and conversely, that it may indicate a larger difference between two groups than humans conventionally think. However, relatively clear differences like the ones between remote dialects seem to be captured by this simple setup.

We point to the two possible future directions, which may appear at odds with each other, to improve on the presented study. One is the use of auto-encoder, which is a popular method amongst the neural net adherents, which may find the latent layers of features in the dataset which are not detectable in the character encodings alone. The other is the extension by heuristics, as in the final 'enhancement' we saw in the preceding section. Humans have their ways to detect differences, and at least to bring the performance to the human level, it could be a more effective route than complicating features, particularly when little data is available. Interpretability is another advantage of this 'feature engineering' route. In general, however, as long as the amount of data is sufficient, deep neural net approaches tend to achieve better performance. Research in both directions, as we see it, is warranted, since both performance and human-friendliness matter in dialect research.

8. Bibliographical References

Cardinal, P., Dehak, N., Zhang, Y., and Glass, J. (2015). Speaker adaptation using the i-vector technique for bottleneck features. In *Proceedings of Interspeech 2015*.

Edgar, R. C. (2010). Search and clustering orders of magnitude faster than blast. *Bioinformatics. 26 (19): 24602461.*, 26:2460–2461.

Han, B., Cook, P., and Baldwin, T. (2011). Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 49th Annual Meeting of ACL*.

Kamper, H., Elsner, M., Jansen, A., and Goldwater, S. (2015). Unsupervised neural network based feature extraction using weak top-down constraints. In *IEEE International Conference on Acoustics, Speech and Signal Processing*.

Lui, M. and Cook, P. (2013). Classifying English documents by national dialect. In *InProceedings of Australasian Language Tchnology Workshop*.

Malmasi, S., Zampieri, M., Ljubešić, N., Nakov, P., Ali, A., and Tiedemann, J. (2016). Discriminating between similar languages and arabic dialect identification: A report on the third dsl shared task. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 1–14, Osaka, Japan, December.

Mannepalli, K., Sastry, P. N., and Suman, M. (2016). Mfcc-gmm based accent recognition system for telugu speech signals. *Int. J. Speech Technol.*, 19(1):87–93, March.

Ranjan, C., Ebrahimi, S., and Paynabar, K. (2016). Sequence graph transform (sgt): A feature extraction function for sequence data mining (extended version).

Saito, I., Kugatsu, S., Asano, H., and Matsuo, Y. (2014). Morphological analysis for Japanese noisy text based on character-level and word-level normalization. In *Proceedings of COLING 2014*.

Scherrer, Y. (2014). Unsupervised adaptation of supervised part-of-speech taggers for closely related languages. In *Proceedings of the 1st Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*.

Sharma, A. and Dhir, R. (2009). A wordset based document clustering algorithm for large dataset. In *International Conference on Methods and Models in Computer Science*.

Williams, J. and Dagli, C. (2017). Twitter language identification of similar languages and dialects without ground truth. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 73–83, Valencia, Spain, April. Association for Computational Linguistics.

Yoshino, K., Hirayama, N., Mori, S., Takahashi, F., Itoyama, K., and Okuno, H. G. (2016). Parallel speech corpora of Japanese dialects. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages –, Paris, France, may. European Language Resources Association (ELRA).

Zaidan, O. F. and Callison-Burch, C. (2013). Arabic dialect identification. *Computational Linguistics*.

9. Language Resource References

(2014). *KAKASI: Kanji Kana Simple Inverter*.

Twitter Inc. (present). *Twitter Stream API*. Twitter Inc.