Automatic Data-Driven Approaches for Evaluating the Phonemic Verbal Fluency Task with Healthy Adults

Hali Lindsay, Nicklas Linz
German Research Center for
Artificial Intelligence (DFKI),
Saarbrücken, Germany
hali.lindsay@dfki.de
nicklas.linz@dfki.de

Johannes Tröger, Jan Alexandersson
German Research Center for
Artificial Intelligence (DFKI),
Saarbrücken, Germany
johannes.troeger@dfki.de
jan.alexandersson@dfki.de

Josef Van Genabith
German Research Center for
Artificial Intelligence (DFKI),
Saarbrücken, Germany
jan.alexandersson@dfki.de

Christoph Kaller
Faculty of Medicine,
Freiburg Brain Imaging Center,
University of Freiburg,
Freiburg, Germany
christoph.kaller@uniklinik-freiburg.de

Abstract

Phonemic Verbal Fluency (PVF) is a cognitive assessment task where a patient is asked to produce words constrained to a given alphabetical letter for a specified time duration. Patient productions are later evaluated based on strategies to reveal crucial diagnostic information by manually scoring results according to predetermined clinical criteria. In this paper, we propose four alternative similarity metrics and evaluate them in a two-fold argument, using the clinical criteria as a baseline. First, we consider the capacity of each metric to model PVF production using a rank-based approach, and then consider the metrics ability to compute finer resolution clinical measures that are indicative of the underlying strategy. Automation of the clinical criteria and proposed metrics are evaluated on PVF performances for 16 letters from 32 healthy German students (n=512). Weighted phonemic edit distance performed best overall for modelling both production and strategy.

1 Introduction

Phonemic Verbal Fluency (PVF) is a standard neuropsychological test that is used to assess cognitive abilities. During this task, a person is asked to produce as many words as possible starting with a given letter in a specified amount of time. Classically, the PVF performance is then scored by counting the total number of unique words produced, however more fine-grained measures of performance (i.e. strategy) have been established to differentiate between multiple pathologies (Gruenewald and Lockhead, 1980). Troyer et al. (Troyer et al., 1997) first proposed a framework for assessing the strategy of a PVF performance: a rule-based system to determine phonemic clusters by manually defining criteria for phonemic similarity (Vonberg et al., 2014). According to this criteria, consecutive words in a production are lumped into categories if they share common first letters (e.g. arm & art), rhyme (e.g. $stand$ & $sand$), share first and last sounds (e.g. sat, $seat$ & $soot$) or are homonyms (e.g. $some$ & sum).

While modelling production strategy (i.e., clustering and switching measures) is crucial for clinical cognitive considerations, the traditional manual approach is subjective and time consuming. There is a clear need for a data-driven automatic approach that addresses these limitations. Novel computational approaches to the analysis of *semantic verbal fluency* (SVF), where patients are asked to produce words based on a semantic cue (e.g. *animals*), could help to overcome the current limitations in PVF analysis (Woods et al., 2016; Linz et al., 2017; Clark et al., 2016; Troeger et al., 2019). The underlying rationale is to use a global similarity metric that is learned from data to derive a notion of relatedness between produced words, which can later be used to determine structures of related clusters as proxy for production strategy.

In the case of SVF, the similarity metric is semantically motivated.

Given the sparse body of research on automatic PVF analysis schemes modelling both production and strategy, further investigation on more sophisticated data-driven modelling approaches to PVF is needed. The goal of this paper is two-fold:

(1) First, we aim to introduce and compare the performance of five different similarity metrics for modelling production of PVF–in cognitively healthy participants–across sixteen letter categories, including an automated version of the current clinical criteria.

(2) Second, we propose a data-driven clustering scheme for determining phonemic clusters as a means of evaluating production strategy. In both experimental conditions, we compare the novel metrics to an implementation of the classic clinical Troyer baseline, described previously, to evaluate performance.

2 Related Work

Little previous research has proposed similar data-driven approaches for PVF evaluation which requires a phonemic similarity metric, respectively. Ryan et al. (Ryan et al., 2013) determined phonemic clusters in PVF tasks using a *phonemic similarity score*, based on edit-distance between phoneme representations from a pronunciation dictionary, and a *common biphone score*, a binary variable encoding the presence of a common initial and/or final biphone. They compared PVF performances (letter F) of martial arts fighters with high and low exposures (according to number of fights) and found significant differences in the groups mean and maximum cluster length for both biphone and phonemic similarity score approaches, and significant differences for the mean pairwise phonemic similarity provided by the common biphone method. This exploratory result demonstrates the potential of automated qualitative PVF analysis in the context of neurocognitive syndromes.

However, this approach does not capture the effect that phonemic properties might influence strategy, e.g. that some phonemes are closer in articulation than others. Previously, authors have proposed methods to weight *edit*-distance between phonemic representations with features reflective of the similarity between phonemes. Fontan et al.(Fontan et al., 2016) used Levenshtein (Levenshtein, 1966) distance between different phonemes, weighted by common features shared between them. Through this, they propose a new metric to evaluate automatic speech recognition systems, that seem to be consistent with human perception. Zampieri et al. (Zampieri and de Amorim, 2014) proposed a metric to enhance target word recovery for spell checking in English where they combined two weighted instances of Levenshtein distance. First, between the edit distance between two words normal spelling is calculated and then between the four digit Soundex code representations, where the Soundex algorithm represents similar sounding words as the same representation. This was combined with clustering techniques to improve spell checking. Similar methods have been used to measure pronunciation differences of dialects in Norwegian where weighted Levenshtein distance using phonetic representations and acoustic features were used with clustering techniques (Heeringa, 2005).

Given this, there is a substantial gap in advancing the state of the art in data-driven modelling of PVF speech output that can be leveraged for clinical applications.

3 Methods

Closing this gap, this section describes four proposed distance metrics for measuring similarity as well as the clinical baseline and details a *rank-cost* evaluation criteria to compare all metrics' ability to model PVF productions. Furthermore, this methodology is used in a second performance evaluation of each metric for modelling clinical clustering and switching strategy based on clusters defined by the affinity propagation clustering algorithm (Frey and Dueck, 2007).

3.1 Modelling Production

3.1.1 Metrics

Levenshtein distance (Levenshtein, 1966) is computed as the number of insertions, deletions and substitutions that are necessary to transform one word into another word. Let d, i and s represent the cost of deletions, insertions and substitutions respectively.

1. *LD*: The Levenshtein distance between the orthographic representation of words

2. *phon*: the Levenshtein distance between phonetic representations, weighted for pho-

netic similarity. Phonological feature vectors are obtained from Epitran using Panphon's database of International Phonetic Alphabet (IPA) symbol features (Mortensen et al., 2016). Each phonetic symbol is represented by a fixed-length vector of integers between -1 and 1 representing the presence (+1), absence (0), or lack (-1) of 21 phonological features. The weighted similarity score for s is the hamming distance between the phonetic vector representations. d and i are held constant at 1.

3. *pos*: Levenshtein distance between phonetic representations, weighted for position in word, d, i and s are set as q, where q is drawn from the exponential distribution at position i, with $\lambda = 0.5$.

4. *sem*: The semantic distance between word vector representation. Semantic representations of word vectors were obtained from the German fastText model (Grave et al., 2018; **?**) and similarity is approximated as the cosine distance between the vectors.

5. *Troyer*: Implementation of Troyer clinical criteria for phonemic clustering (Troyer et al., 1997). Values were calculated by (1) string matching the first or last 2 letters, (2) matching the first two sounds of phonetically transcribed words, (3) for rhyming, matching the last two sounds of phonetically transcribed words and (4) for homophones, matching phonetic transcriptions of the whole word. Each criteria was weighted as 1 and the sum of criteria present was used as a score. The max score was a 4 and the lowest 0. Words with equivalent scores where sorted alphabetically.

Phonetic transcriptions were obtained with Epitran, a python library that translates orthographic to phonetic representations (Mortensen et al., 2018).

For each letter category, c, in our data set a vocabulary of the set of all words produced, V_c, is constructed. The vocabulary V_c has length N. For each of the described similarity metrics f, a table of size NxN is created where the similarity between every word in vocabulary is calculated. The result is a square, symmetric similarity matrix, S_c, for each metric.

3.1.2 Evaluation

Difference of scale for each of the metrics renders direct comparison impossible, therefore performance of the metrics is evaluated via ranking tables.

For each similarity matrix of a letter category S_c, a list is generated for every word in the vocabulary, V_c, of the most similar to the least similar as determined by the metric f. To formalize this, a rank table T is created for every word w in each letter vocabulary V_c.

Once all tables are populated, the rank cost of the PVF samples RC_f are calculated by c for each f. Given a production $P = w_1...w_n$, a metric f and ranking tables for each word $T^f_{w_1}...T^f_{w_n}$ the rank cost of P, given f, is determined as

$$RC_f(P) = \frac{\sum_{j=1}^{n-1} T^f_{w_j}[w_{j+1}]}{n-1}$$

Using rank based comparison is motivated by a two arguments. First, ranking makes different similarity metrics comparable, by rendering issues of scale irrelevant while preserving the individual metrics outcome. Second, the resulting RC_f can be interpreted directly as the offset of the mean rank, when used for predicting the next word from our vocabulary. The similarity metric f which is better at modelling production will have a lower RC_f.

3.2 Modelling Strategy

3.2.1 Metrics

After modelling production, it is crucial to consider that the clinical *Troyer* metric is not a method of modelling production, but rather a clustering strategy to explore the underlying cognitive process of this clinical task. Taking this into account, the following methodology aims to compare each metric's ability to model the underlying strategies of the PVF task.

Affinity Propagation Clustering (AP clustering) is a clustering algorithm based on each point in a data set—in this application, the similarity matrix S_c for each metric f—passing messages simultaneously through two matrices, representing either responsibility or availability. The end result is an emergence of data points—or words from V_c—that are considered exemplars, having high responsibility, while remaining points are then grouped around the exemplars to create clusters,

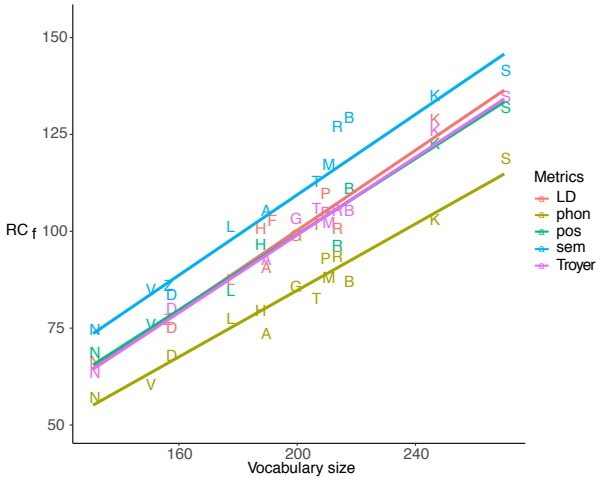

Figure 1: Median RC_f for each letter and method f as a function of vocabulary size. Different f are indicated by color. Lines indicate fit of a linear model.

or better suited by availability (Frey and Dueck, 2007). A unique point of AP clustering is that the number of clusters is not predefined, but emerges from the data. This concept lends itself naturally to the idea of clustering in PVF, as exemplars can be seen as the general topic that is being searched for during the production.

To apply this to the data, for every letter category c, the generated similarity matrix S_c for each metric f is used to create a set of clusters as determined by AP clustering algorithm. The resulting clusters are then saved and applied to each production in the data set to consider the strategy estimated by each metric. Consecutive words in each participant production are compared to see if they belong in a cluster as determined by each similarity metric.

For example, if a participant was given the letter category C, they might produce the following:

cat, crab, crawl, crib, cash, cache

The clusters generated from a selection of the similarity metrics using the AP clustering algorithm to cluster the PVF performance would yield the following, where words within a set of brackets indicate a computed cluster:

Troyer: [cat], [crab, crawl, crib], [cash, cache]
sem: [cat, crab], [crawl, crib], [cash], [cache]

3.2.2 Evaluation

The quality of the AP clustering technique on this task is evaluated using the silhouette coefficient. This measure is ideal as it does not require a ground truth. This measure looks at the fit of a cluster by considering if every point is in its closest cluster, or if another cluster would be more suitable. Each point in the dataset is considered. First, the average distance between the chosen point and all points in its own cluster ($distance_{cohesion}$) is calculated. Then, the average distance between the same point and all points in next nearest cluster is calculated ($distance_{separation}$).

$$\frac{distance_{separation} - distance_{cohesion}}{max(distance_{separation}, distance_{cohesion})}$$

The silhouette coefficient is bounded from -1 to 1, where positive values indicate higher quality clusters and negative values typically indicate that a point has been incorrectly clustered (Rousseeuw, 1987).

The ability of the metrics to model strategy is evaluated by looking at the average rank cost within clusters as well as the average rank cost between clusters, or switches. The rank cost tables created previously are used to calculate this respectively.

The average rank cost of clusters is calculated by looking at the rank cost of transitions between words in each cluster and normalized by the number of transitions in a cluster.

The average rank cost of switches in a production is calculated by summing the rank costs of transitions between cluster boundaries and normalizing by the number of switch transitions.

Metrics with a lower average rank cost within clusters and higher average rank cost of switching are seen to better model strategy.

4 Experiment 1: Modelling Production

For the first experiment, one minute PVF performances of 32 German students (9 male, 23 female; Age 22.88) from 16 different letter categories (i.e. *A, B, D, F, G, H, K, L, M, N, P, R, S, T, V, Z*) were collected. These were manually transcribed on a word level into sequences of correct responses. Words were converted into phoneme (IPA) representations using the python *epitran*[1] package. For each letter category c, a vocabulary V_c was constructed to calculate the RC_f of each sample as described in Section 4.

Statistical analysis was performed using R (software version 3.4.0). Performance of metrics over all letters was examined with a linear mixed

[1] https://github.com/dmort27/epitran

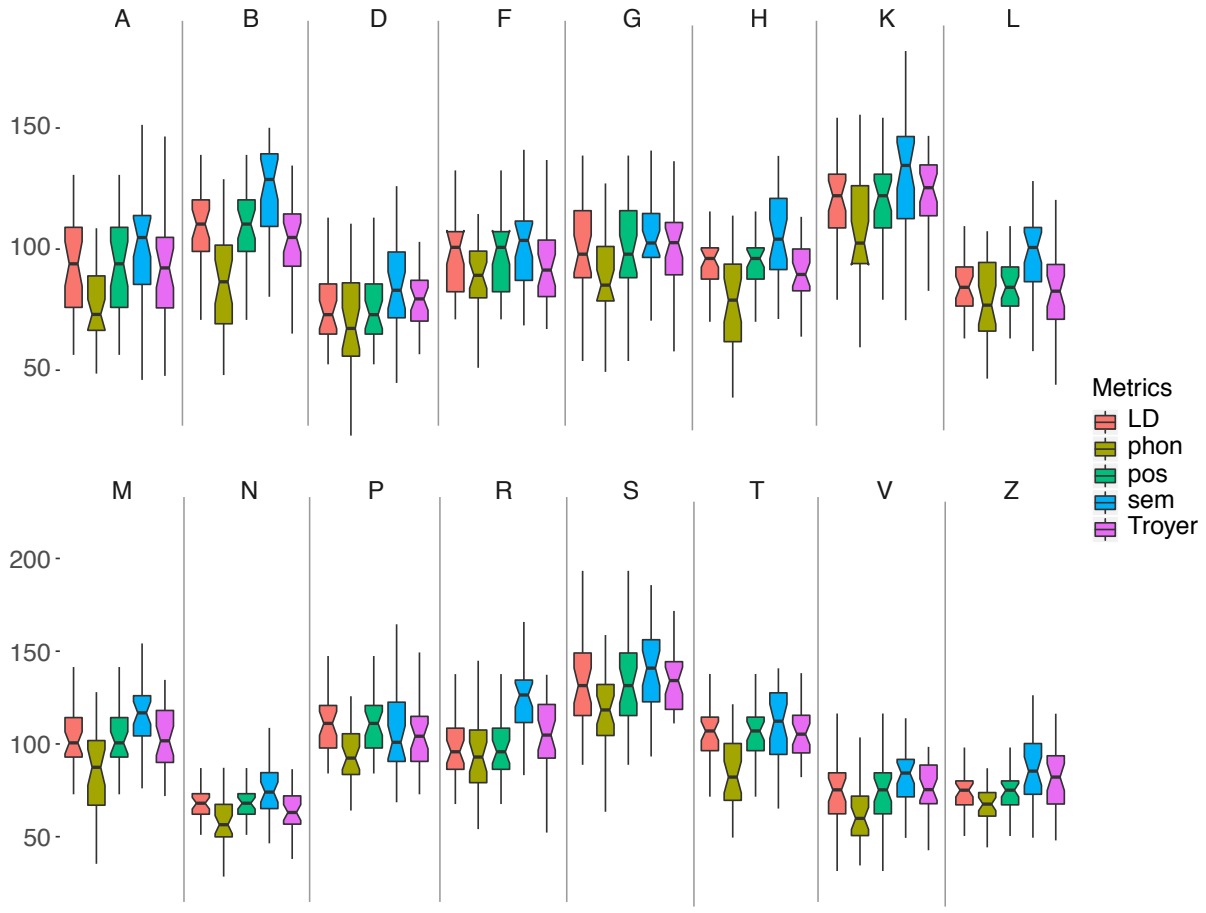

Figure 2: Comparison of RC_f values for distance metrics f and letter categories. Each boxplot represents one letter category and contains results from the five distance metrics defined in Section 3.1. In the case of black and white prints, metrics for each letter category match the legend from top to bottom as left to right.

effects analysis using the *lme4* (Bates et al., 2014) package. Each RC_f was modelled as a single data point and letter and metric were represented as fixed effects. The participant identifier was modelled as a random intercept.

5 Experiment 2: Modelling Strategy

The affinity propagation clustering algorithm was implemented in python from *scikit-learn* framework (Pedregosa et al., 2011). The same parameters were used to determine all models. The preference parameter serves as an indicator of how fit a word in the vocabulary is to be an exemplar, higher values indicate that it is more likely where as lower values indicate that it is less likely. This also influences the number of clusters produced, where higher preference values lead to more clusters and lower preference values lead to fewer cluster. The preference parameter was set for each word in the vocabulary as the Zipf word frequency as determined by the python wordfreq package (Speer et al., 2018). The zipf word frequency represents the frequency of the word in a large, in this case German, corpus on a 'human-friendly' scale. The result is a value between 1.0 and 8.0, where the larger the value, the more frequent the word is in the language. The goal of using the word frequency during clustering is to give a high exemplar weight to more frequent words to make the clusters relevant to the PVF production task. The remaining parameters were left at their default values; the damping factor was set to 0.5 and convergence iteration rate at 200. Each previously computed similarity matrix S_c was used as an input to generate clusters for each metric f.

The average rank cost of clusters in a production was computed as described in 3.2.2.

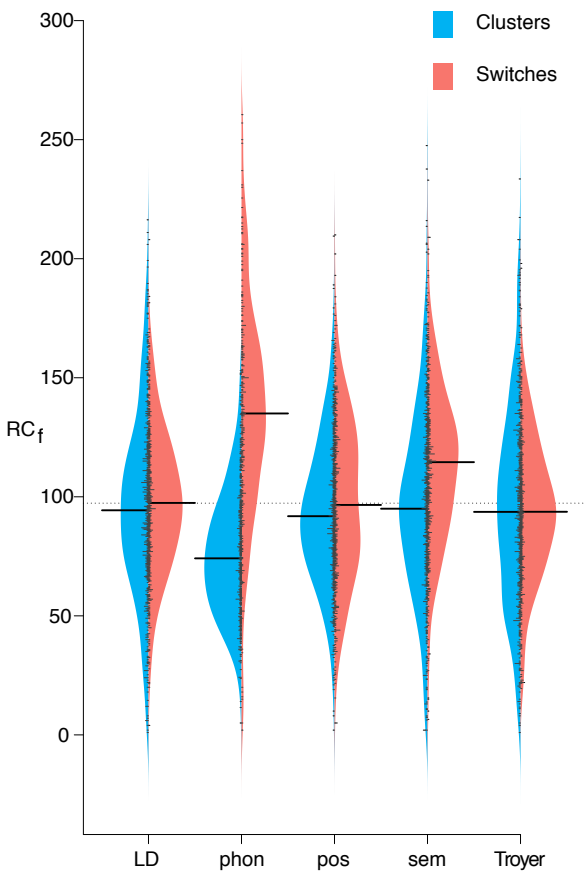

Figure 3: Beanplots comparing the distribution of average rank cost of clustering and switching across all letter categories, by metric. The left distribution is for clustering and the right distribution is for switching. The long bar in the distribution represents the median.

6 Results

6.1 Experiment 1

Results are displayed in Figure 2, where a better fit is indicated by lower RC_f. One boxplot is shown for each letter category. The threshold for rejecting a null hypothesis and determining statistical significance is set at 0.05 for all tests performed.

The linear models were created as described in 3.1.2 and revealed that RC_f values were significantly lower for the *phon* and significantly greater for the *sem* metric. Performances varied across letter categories with the lowest overall RC_f values being observed for the letter *N* and the highest for *S*.

6.2 Experiment 2

Evaluation of cluster quality as produced by the AP clustering algorithm is monitored via their silhouette coefficients as described in section 3.2.2 and are shown in Table 1.

The highest quality clusters were produced by the *phon* metric. The *pos* metric had the second highest quality on average. The remaining metrics all produced relatively close values for all letter categories with *Troyer* performing slightly better than *LD* and *sem*. Overall, all metrics on average produced positive cluster values.

LD	phon	pos	sem	Troyer
0.025	0.738	0.330	0.083	0.170

Table 1: silhouette coefficients

Figure 3 uses beanplots to compare each metric by the distribution of average rank cost within a cluster and the average rank cost of switches. *Phon* had a much lower average rank cost within clusters where as all other metrics were relatively equal, with *Troyer* having slightly lower than LD. Sem had the highest average cluster rank cost.

For each metric, a paired-samples t-test was conducted to compare average RC_f, aggregated across letter categories, between clustering and switching conditions. There were significant differences in average rank cost for clustering and switching for *phon* ($t(222)=-20.17$, $p<0.05$), *sem* ($t(222)=3.69$, $p<0.05$) and *pos* ($t(222)=-2.372$, $p<0.05$). No significant differences were found for the metrics LD or Troyer.

7 Discussion

For modelling the entire production, *phon* outperformed the *troyer* and *LD* metrics in every letter category, showing an improvement from our baseline measurements. Overall, the metric that best modeled the data based on the ranked cost evaluation was *phon*. The semantic similarity measure *sem* had the highest average rank cost across all letter categories, leading us to believe that the task as a whole is not semantically motivated.

For modelling strategy based on clustering and switching, the phonetically weight edit distance *phon* continued to have the highest quality clusters as indicated by a low rank cost across all letter categories. This metric also best modelled the switching procedure between clusters as indicated by a high rank cost. In addition,

While the semantically motivated *sem* metric performed poorly on modelling the overall production it was able to capture the relationship of clustering strategy, albeit not as well as *phon*. This

could be due to the lower quality of clusters produced by the sem metric, as determined by the silhouette coefficient, however the overall score is within a reasonable range. Another consideration is that the phonemic task has little semantic underlying notions for producing clusters and phonemically derived measures are more suited to the task. There is also a possibility that within phonemic verbal fluency there are phonemic and semantic strategies that motivate clustering and switching. For example, a cluster of the words "grandmother", "grandfather", and "grandstand" would be both semantically and phonemically motivated.

8 Conclusion

This paper compared different similarity metrics for their ability to model production in PVF for multiple letter categories. The proposed *phon* approaches significantly outperformed the simple *LD* baseline and automated *troyer* methods for both modelling production and strategy. Surprisingly, the *sem* metric performed poorly in comparison to all other metrics when modelling the entire production sequence, but was able to capture the notion of underlying strategies of clustering and switching.

Further development of the newly proposed metrics should be continued by tuning parameters for AP clustering per evaluated metric to achieve higher quality clusters rather than the uniform configurations demonstrated in this paper. Further investigations could also combine semantic and phonemic methods by classifying clusters as being either semantically motivated or phonemically motivated. The next step in this line of research would be to apply these new PVF techniques in a clinical application and evaluate the effectiveness of these features to distinguish between different pathological groups. Similar evaluations should be conducted for other languages, since results may vary due to phonemic differences.

References

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2014. Fitting linear mixed-effects models using lme4. *arXiv preprint arXiv:1406.5823*.

D. G. Clark, P. M. McLaughlin, E. Woo, K. Hwang, S. Hurtz, L. Ramirez, J. Eastman, R. M. Dukes, P. Kapur, T. P. DeRamus, and L. G. Apostolova. 2016. Novel verbal fluency scores and structural brain imaging for prediction of cognitive outcome in mild cognitive impairment. *Alzheimers Dement (Amst)*, 2:113–122.

Lionel Fontan, Isabelle Ferrané, Jérôme Farinas, Julien Pinquier, and Xavier Aumont. 2016. Using phonologically weighted levenshtein distances for the prediction of microscopic intelligibility.

Brendan J. Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*, 315(5814):972–976.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Paul J Gruenewald and Gregory R Lockhead. 1980. The Free Recall of Category Examples. *Journal of Experimental Psychology: Human Learning and Memory*, 6:225–240.

Wilbert Heeringa. 2005. Measuring dialect pronunciation differences using levenshtein distance. *Zeitschrift fr Dialektologie und Linguistik*, pages 205–208.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals.

Nicklas Linz, Johannes Tröger, Jan Alexandersson, and Alexandra König. 2017. Using Neural Word Embeddings in the Analysis of the Clinical Semantic Verbal Fluency Task. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS)*.

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori S. Levin. 2016. Panphon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484. ACL.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peter Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis.

James O Ryan, Serguei VS Pakhomov, Susan E Marino, Charles Bernick, and Sarah Banks. 2013. Computerized analysis of a verbal fluency test. In *Proceedings of ACL*, pages 884–889.

Robyn Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. 2018. Luminosoinsight/wordfreq: v2.2.

Johannes Troeger, Nicklas Linz, Alexandra Knig, Philippe Robert, Jan Alexandersson, Jessica Peter, and Jutta Kray. 2019. Exploitation vs. explorationcomputational temporal and semantic analysis explains semantic verbal fluency impairment in alzheimer's disease. *Neuropsychologia*.

Angela K Troyer, Morris Moscovitch, and Gordon Winocur. 1997. Clustering and Switching as Two Components of Verbal Fluency: Evidence From Younger and Older Healthy Adults. *Neuropsychology*, 11(1):138–146.

Isabelle Vonberg, Felicitas Ehlen, Ortwin Fromm, and Fabian Klostermann. 2014. The absoluteness of semantic processing: Lessons from the analysis of temporal clusters in phonemic verbal fluency. 9:e115846.

David L. Woods, John M. Wyma, Timothy J. Herron, and E. William Yund. 2016. Computerized Analysis of Verbal Fluency: Normative Data and the Effects of Repeated Testing, Simulated Malingering, and Traumatic Brain Injury. *PLOS ONE*, 11(12):1–37.

Marcos Zampieri and Renato Cordeiro de Amorim. 2014. Between sound and spelling: Combining phonetics and clustering algorithms to improve target word recovery. In *PolTAL*.