# The Social and the Neural Network: How to Make Natural Language Processing about People again

**Dirk Hovy**
Bocconi University
Via Roberto Sarfatti 25
20136 Milan, MI, Italy
mail@dirkhovy.com

## Abstract

Over the years, natural language processing has increasingly focused on tasks that can be solved by statistical models, but ignored the social aspects of language. These limitations are in large part due to historically available data and the limitations of the models, but have narrowed our focus and biased the tools demographically. However, with the increased availability of data sets including sociodemographic information and more expressive (neural) models, we have the opportunity to address both issues. I argue that this combination can broaden the focus of NLP to solve a whole new range of tasks, enable us to generate novel linguistic insights, and provide fairer tools for everyone.

## 1 Introduction

Up until the 1970s, economic theory assumed that people make economic decisions with their own best interest in mind, and based on the full available information. This was a useful assumption, which allowed researchers to model people, firms, and markets as statistical linear models of the form $y = \mathbf{w}^T\mathbf{x}$, to test existing theories and to generate new insights. The seminal work by Tversky and Kahneman (1973), however, showed that this assumption was wrong: they demonstrated experimentally that again and again, people would make economic decisions that were not in their best interest, even with the full available knowledge, but instead relied on biases and heuristics. This did not mean that the linear models were useless they were useful abstractions. It did show, however, that there was more to the subject, and that it was fundamentally about people. Incorporating people's behavior opened up economics to new insights, and even established a completely new field, behavioral economics.

Up until the 1990s, NLP was largely based on applying heuristics based on linguistic theory. However, in the 1990s, the field underwent a "statistical revolution": It turned out that statistical linear models of the form $y = \mathbf{w}^T\mathbf{x}$ were more robust, accurate, and reliable in extracting linguistic information from text than linguistic heuristics were. This was a useful insight, which enabled us to solve a number of tasks. However, as a consequence, the field focused more and more on tasks that *could* be solved with these models, and moved away from tasks that could not. While this approach enabled a number of breakthroughs, it also increasingly narrowed the focus of the field, in what could be called "streetlamp science": much like the person searching for their keys under the light of the streetlamp (rather than where they lost them), NLP has continued to search for tasks that could be solved by the statistical models we have, rather than the ones that could help us understand the underpinnings of language.

This shift to the streetlamp and away from the social aspects of language has had two practical consequences: it ignored a whole host of applications that are more difficult to model, and it biased our tools. Language is about much more than information: language is used by people to communicate with other people, to establish social order, to convince, entertain, and achieve a whole host of other communicative goals, but also to signal membership in a social group.

The latter is most obvious in teenagers, who become linguistically creative to distinguish themselves from their parents. For most other groups, the process is much less obvious and often subconscious, but all people use language to mark their membership in a variety of demographic groups:

these groups range from gender to region, social class, ethnicity, and occupation. This property of language has been used in NLP to predict those demographic labels from text in author-attribute prediction tasks (Rosenthal and McKeown, 2011; Nguyen et al., 2011; Alowibdi et al., 2013; Ciot et al., 2013; Liu and Ruths, 2013; Volkova et al., 2014, 2015; Plank and Hovy, 2015; Preoţiuc-Pietro et al., 2015a,b, inter alia).

However, demographics also affect NLP beyond their use as prediction target. Demographic bias in the training data can severely distort the performance of our tools (Jørgensen et al., 2015; Hovy and Søgaard, 2015; Zhao et al., 2017), while accounting for demographic factors can actually improve performance in a variety of tasks (Volkova et al., 2013; Hovy, 2015; Lynn et al., 2017; Yang and Eisenstein, 2017; Benton et al., 2017). In order to move forward as a field, we will have to follow two strands of research: 1) we need to identify the specific demographic factors that do have an influence on NLP models (on bias and performance), and 2) based on this knowledge, we need to develop models that account for demographics to improve performance while preventing bias.

In this position paper, I argue that the recent abundance of demographically rich data sets and complex neural architectures allows us to break out of streetlamp science and to explore those two strands of demographically-based research. This shift will enable a host of new applications that make socio-demographic aspects an integral part of language. I highlight several neural network architectures and procedures that show promise to achieve these goals, and provide some experimental results in applying them.

## 2 Neural models for sociolinguistic insights

### 2.1 Representation Learning

Word embeddings have been shown to be effective as input in a variety of NLP tasks, because they are able to capture similarities along a large number of latent dimensions in the data. If language is indeed a signal for socio-demographic factors, it makes sense to assume that these socio-demographic factors are captured as latent dimensions in continuous word representations.

Indeed, Bamman et al. (2014) have shown that neural representations can be used to capture extra-linguistic information about geographic variation, by adding US-state specific representations to general word word embeddings. The resulting vectors capture regional factors, such as the nearest neighbors for landmarks, parks, and sports teams. In the same vein, Hovy (2015) showed that if word embeddings are learned on corpora that have been explicitly based on certain demographic attributes, they capture these underlying factors to influence performance of text-classification tasks sensitive to them.

It is easy to extend this concept to a popular and widely available representation-learning tool, `paragraph2vec` (Le and Mikolov, 2014). `Paragraph2vec`, similar to `word2vec` (Mikolov et al., 2013), learns embeddings through back-propagation of the input (and output) representations in a simple prediction task. Depending on the precise architecture, we either have document labels as inputs and words as output (DBOW), or words and documents as input and words as output (DM).

Instead of separating out different sub-corpora or including modifiers to the general word embeddings, though, we can exploit the unsupervised learning setup of the model, by using socio-demographic attributes (if known) as document labels (rather than unique document identifiers). Crucially, we can provide as many labels as we want for each document (see Table 1 for examples of this).

Through the training process, latent characteristics of the document labels are reflected in the learned word embeddings, while the embeddings of the demographic labels reflect the words most closely associated with them.

| TEXT | LABELS |
|---|---|
| I had a lovely experience with them | F, 60, ID00014 |
| ... | |
| Compared lots of prices and ended up with them. Good value for money | ID16457 |
| ... | |
| Exactly the product I wanted. Good price and speedy delivery. | M, ID243534 |

Table 1: Example reviews with different amounts of available labels

As a result, we have representations of the word, document, and population-level. The unique document identifiers allow us to represent each training instance as a vector. The socio-demographic labels, on the other hand, are not unique, but shared among potentially many instances.

In the `gensim` implementation of `paragraph2vec`, both word and document-label embeddings are projected into the same high-dimensional space. We can compare them using cosine-similarity and nearest neighbors.

This allows us to qualitatively examine four comparisons:

1. words to words: similar to `word2vec`, this allows us to find words with similar meanings, i.e., words that occur in a similar context. In addition, these words representations are conditioned on the socio-demographic factors, though.

2. words to document labels and

3. document labels to words: this allows us to find the $n$ words best describing a document label, or the $n$ document labels most closely associated to a word

4. document labels to document labels: this allows us to find similarities between socio-demographic factors.

In addition, we can use clustering algorithms on the word and document representations to identify

1. topic-like structures (when clustering on the word representations)

2. extra-linguistic correlations (when clustering on the document representations)

I will illustrate the four different comparisons in a study on the data from Hovy et al. (2015)[1] below, as well as the two clustering solutions. I use English reviews labeled with the age, gender, and location of the author. Note that in the setup described here, we do *not* need to have all the information for all instances! We can use evidence from partial labeling to exploit a larger sample.

Note that the methodology described here is by no means limited to socio-demographic factors, but can be applied to other variables of interest. The advantage of this method is that it requires no new model, can be used on a wide variety of input sources and problems, and yields interpretable results. We provide an implementation of the entire pipeline suit (representation learning, clustering) as a Python implementation on github: `https://github.com/dirkhovy/PEOPLES`.

[1] `https://bitbucket.org/lowlands/release/src/fd60e8b4fbb1/WWW2015/`

## 2.2 Experimental Results

I preprocess the data to remove stop words and function words, replace numbers with 0s, lowercase all words and lemmatize them. I also concatenate collocations with an underscore to form a single item. This reduces the amount of noise in the data. As labels, I use the seven age decades, as well as the two genders present in the data. Overall, this results in slightly over 2M instances. See Table 1 for examples.

I run the model for 100 iterations, following the settings described in (Lau and Baldwin, 2016), with the embedding dimensions to 300, window size to 15, minimum frequency to 10, negative samples to 5, downsampling to 0.00001.

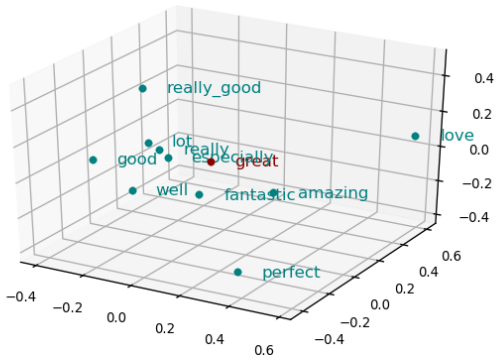


Figure 1: 10 nearest neighbors of *great* in 3-dimensional space.

**Comparing words to each other** The effect of the modeling process is that semantically similar words get closer in embedding space. The 10 nearest neighbors when querying for the word *great* are *well, fantastic, amazing, really_good, good, really, lot, perfect, especially*, and *love* (see Figure 1 for a graphical depiction). This is not new or surprising, but I will show further results building on this in subsequent sections.

**Comparing words to labels** We can use each demographic label vector and find the closest

words around them. This gives us descriptors of the labels.

10: yesstyle, cd_key, game, cjs_cd_keys, cjs
20: ever, never, today, nothing, anything
30: nothing, actually, complain, today, even
40: sort, company, nothing, fault, -PRON-
50: sort, advise, fault, realise, problem
60: telephone, problem, firm, certainly, sort
70: could_find, certainly, good, problem, certainly_use
F: brilliant, lovely, fab, really_pleased, delighted
M: fault, sort, round, good, first_class

We can also use the well-known vector arithmetics that allow us to subtract and add vectors from each other. Using the example word from the previous paragraph, *great*, but adding and subtracting demographic label representations in the calculation, we can compute

$$great - MALE + FEMALE$$

and

$$great - FEMALE + MALE$$

to see which words women and men, respectively, use with or for *great*.

The first calculation give us *fab*, *fabulous*, *lovely*, *love*, *wonderful*, *really_pleased*, *fantastic*, *brilliant*, *amazing*, and *thrill* for women and *guy*, *decent*, *good*, *top_notch*, *couple*, *new*, *well*, *gear*, *get_good*, and *awesome* for men.

Such knowledge is interesting with respect to sociodemographic studies, but can have practical applications: Reddy and Knight (2016) have shown how gender can be obfuscated online by replacing particularly "male" or "female" words with a neutral or even opposite counterpart. The approach shown here based on vector arithmetics is a possible simple alternative.

**Comparing labels to labels** Comparing labels to each other is again very similar to the situation we have seen above for words. In the present study, this comparison is less interesting (though we can for example see which age groups are more or less similar to each other, see Figure 2).

However, we will exploit this attribute in the next section (2.3), were we explicitly compare labels to each other.

**Clustering** Clustering the word representations with $k$-means gives us a number of centroids in the embedding space, which we

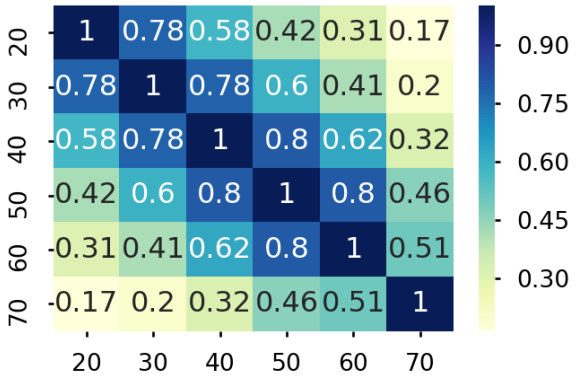


Figure 2: Cosine similarity of vector representations of age groups.

can again characterize by their closest words. For 10 clusters, we see: TROPHIES: trophiesplusmedal, trophy, medal, trophy_store, good_product_good_price_excellent_delivery_time
CUSTOMER SERVICE: confirm_-PRON-_account, wojtek, activate, first_time_order_part_geek, frustrated
MOBILE PHONES: mazuma, send_-PRON-_phone, send_phone, great_service_would_use_recommend_friend, mazuma_mobile
TASTE: taste, flavour, delicious, protein, tasty
CARS: mechanic, bmw, partsgeek, -PRON-_vehicle, -PRON-_car
GLASSES: pair_glass, optician, -PRON-_glass, -PRON-_prescription, glass
SHIPPING: excellent_service_order_arrive_day, first_class_service_would_recommend_-PRON-_friend, guitar, good_service_fast_delivery_excellent_product, reliable_service_prompt
SERVICE: excellent_service_prompt_delivery_good_price, refuse, tell, apparently, akinika
MISC: srv, hendrix, marvin, bankcard, irrational
TRAVEL: hotel, airport, flight, -PRON-_flight, -PRON-_trip

### 2.3 Including external knowledge

The last section showed how the learned representations are useful for a variety of qualitative analysis. However, their utility can be improved by leveraging existing outside-information that we did not include as document labels in the training process of the model, either because it was un-

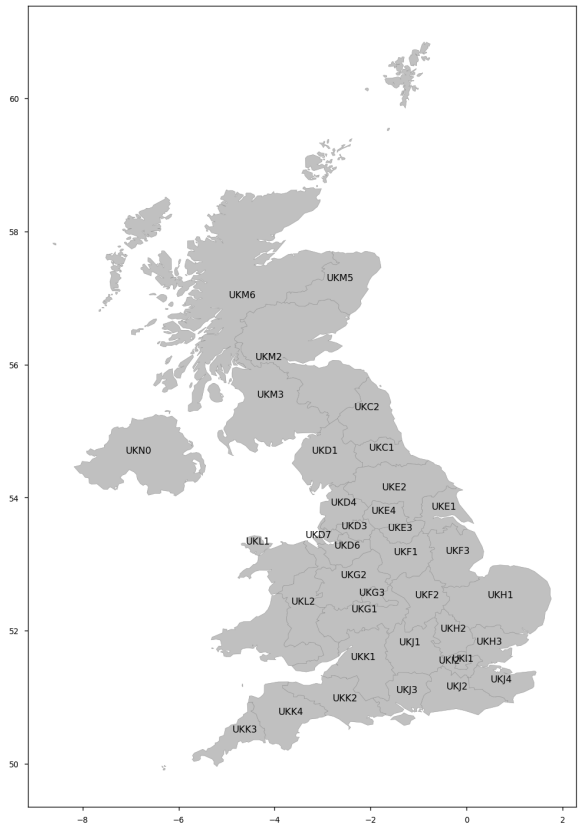


Figure 3: NUTS regions for the UK.

available, could not be incorporated (for example continuous values), or because it serves a different, tasks-specific purpose (whereas the embeddings are general-purpose). Examples of this include knowledge about word or document-label similarities based on some external source.

I provide an intuitive example of these techniques in a setup where we investigate the geographic distribution of terms, and their ability to define larger dialect regions. The input to our model are the geo-tagged tweets and Twitter profile texts (short self-descriptions) from 118K users in the UK, labeled with the statistical geographic region (NUTS2, similar in size to a county) they originated from (see Figure 3). I use the same preprocessing and modeling procedure as before, but in this case only use the regions as document labels.

Due to the nature of online conversations, the most indicative words for each region are typically cities and places in that region (see examples below).[2] An interesting exception to this rule are the regions in Scotland: here, we see several gaelic words among the top-3 (*wee*, *aye*, *nae*).

UKC1: durham, mam, middlesbrough
UKC2: newcastle, sunderland, nufc
UKD1: cumbria, carlisle, workington
UKD3: manchester, mufc, mcfc
UKD4: blackpool, preston, lancashire
UKD6: cheshire, warrington, chester
UKD7: liverpool, everton, lfc
UKE1: hull, hcafc, notohulltigers
UKE2: york, scarborough, harrogate
UKE3: sheffield, reyt, swfc
UKE4: leeds, leed, bradford
UKF1: nottingham, derby, nffc
UKF2: leicester, lcfc, northampton
UKF3: lincoln, lincolnshire, superbull
UKG1: worcester, nuneaton, hereford
UKG2: stoke, coverdrives, nymets
UKG3: birmingham, west_midlands, coventry
UKH1: norwich, suffolk, ipswich
UKH2: hertfordshire, watford, albans
UKH3: essex, colchester, southend
UKI1: london, w/, pic
UKI2: london, loool, lool
UKJ1: oxford, need, find
UKJ2: brighton, sussex, surrey
UKJ3: southampton, portsmouth, hampshire
UKJ4: kent, canterbury, maidstone
UKK1: bristol, bath, cheltenham
UKK2: bournemouth, somerset, dorset
UKK3: cornwall, cornish, truro
UKK4: plymouth, exeter, devon
UKL1: swansea, welsh, wales
UKL2: cardiff, wales, welsh
UKM2: edinburgh, wee, aye
UKM3: glasgow, wee, celtic
UKM5: aberdeen, nae, imorn
UKM6: inverness, caley, rockness
UKN0: belfast, ulster, irish

**Clustering with structure** We can cluster the document labels with agglomerative clustering. This clustering algorithm starts with each region vector in its own cluster, and recursively merges pairs until we have reached the required number of clusters. The pairs to merge are chosen as to minimize the increase in linkage distance. While a variety of distance measures exist, the most commonly used (and empirically most useful) is Ward

[2] Eisenstein et al. (2010) have therefore approached dialects as regionally distributed topics, and Salehi et al. (2017) showed that using such regional terms makes individuals more likely to be correctly geo-located.

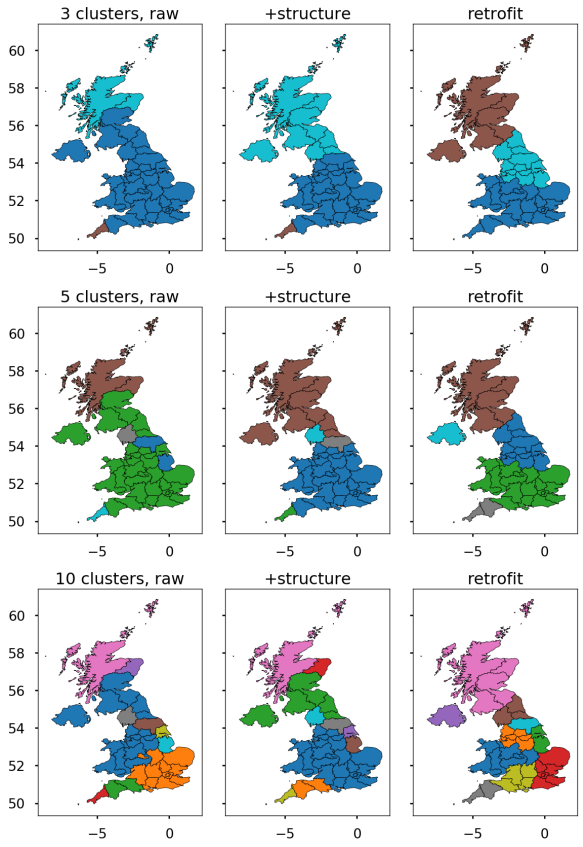


Figure 4: Effect of structure and retrofitting on clustering region embeddings.

linkage, which minimizes the new cluster's variance.

However, while the resulting solutions are stable across runs (as opposed to $k$-means, which is stochastic), they favor creating small new clusters, before breaking up larger groups. The effect can be seen in the leftmost column of Figure 4: one large area dominates, with some smaller regions scattered about. For 5 and 10 clusters, we also see discontinuities.

The algorithm can be enhanced with structure, by providing a connectivity matrix for the data points (i.e., either a floating point similarity or a binary adjacency), which is used to select cluster pairs during the merging process. This structure allows us to infuse the representations with additional knowledge.

Using a binary adjacency matrix over neighboring regions adds additional geographic information to the clustering process, which before was only based on linguistic similarity. We see larger dialect areas emerge, and no more discontinuous dialect areas (center column in Figure 4).

Note that we are not restricted to binary adjacency: if we were comparing points rather than regions (say, individual cities), we could instead use a similarity matrix with the inverse distance between cities (closer cities are therefore merged before more distant cities). This structure lets us express continuous values, which are impossible to include in the learning setup of `doc2vec`.

**Retrofitting** Faruqui et al. (2015) introduced the concept of retrofitting vectors to external dictionaries. This allows us to adjust the positions of the vectors according to categorical outside information.

Here, we convert the adjacency matrix used before into an external dictionary that lists for each region its directly neighboring regions. Retrofitting the region representations under this dictionary forces the representations of adjacent regions to become more similar in vector space. Retrofitting therefore allows us to bring external, geographic knowledge to bear that could not be encoded in the representation learning process.

Clustering the retrofit region embeddings (rightmost column in Figure 4) results in continuous, large dialect areas.[3]

Similarly, we could derive a dictionary that lists for each word all other words observed in the same regions. This second dictionary could be used to adjust the word embeddings along the same lines as the region representations.

## 3 Debiasing and other applications

The previous sections have outlined how representation learning allows us to encode socio-demographic attributes in word and document representations. I have shown a number of qualitative studies that allow us to explore the effect of demographics on language. This is useful in discovering demographic traits,

However, it has been shown that knowledge of socio-demographic variables can improve a variety of NLP classification tasks, either by using them as input features (Volkova et al., 2013), or by conditioning embeddings on various demographic factors (Hovy, 2015). This theme was extended on by (Lynn et al., 2017), who show that user-demographics can be incorporated in a variety of ways, including from predicted labels. Benton

[3] Using structure when clustering these retrofit vectors has no effect, since the information is already encoded in the vectors, so the adjacency matrix adds no additional information.

et al. (2017) show how multitask-learning allows us to include demographic information in prediction tasks by making one of the auxiliary tasks a user-attribute prediction task. Especially in cases where the main task is strongly correlated with the prediction target, MTL can be a promising neural architecture to improve performance. Yang and Eisenstein (2017) have shown another way in which external knowledge about social structures can be incorporated into neural architectures (via attention), to improve prediction accuracy.

At the same time, demographic factors do create a demographic bias in the training data that influences NLP tools like POS taggers (Jørgensen et al., 2015; Hovy and Søgaard, 2015), leading to possible exclusion of under-represented demographic groups (Hovy and Spruit, 2016). Current methods, however, still fail to explicitly account for such biases, and can in fact even increase the demographic bias (Zhao et al., 2017). While it is possible to counter-act this bias, it requires our specific attention. Adversarial learning techniques could present a way to address this problem directly in a neural architecture, similarly to its use in domain-adaptation. This is an area that deserves special attention, if we want to use NLP for social good, and counteract the prevailing problem of biased machine learning.

## 4 Conclusion

In this position paper, I have argued that language is fundamentally about people, but that we have de-emphasized this aspect in NLP. However, with the increased availability of demographically-rich data sets and neural network methods, I argue that we can re-incorporate socio-demographic factors into our models. This will both improve performance, reduce bias, and open up new applications, especially in dialogue, chat, and interactive systems. I show the basic usefulness of representation learning for qualitative socio-demographic studies, and demonstrate several ways that allow us to include further outside knowledge into the representations. In the future, we need to better understand the exact influence of various demographic factors on our models, and develop ways to deal with them. Adversarial learning, multitask learning, attention, and representation learning currently look like promising instruments to achieve these goals.



## References

Jalal S Alowibdi, Ugo A Buy, and Philip Yu. 2013. Empirical evaluation of profile characteristics for gender classification on twitter. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*. IEEE, volume 1, pages 365–369.

David Bamman, Chris Dyer, and Noah A Smith. 2014. Distributed representations of geographically situated language. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. volume 2, pages 828–834.

Adrian Benton, Margaret Mitchell, and Dirk Hovy. 2017. Multitask learning for mental health conditions with limited social media data. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. volume 1, pages 152–162.

Morgane Ciot, Morgan Sonderegger, and Derek Ruths. 2013. Gender Inference of Twitter Users in Non-English Contexts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Wash*. pages 18–21.

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. 2010. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1277–1287.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 1606–1615.

Dirk Hovy. 2015. Demographic factors improve classification performance. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. volume 1, pages 752–762.

Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. User review-sites as a source for large-scale sociolinguistic studies. In *Proceedings of WWW*.

Dirk Hovy and Anders Søgaard. 2015. Tagging performance correlates with author age. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

Dirk Hovy and Shannon L Spruit. 2016. The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. volume 2, pages 591–598.

Anna Jørgensen, Dirk Hovy, and Anders Søgaard. 2015. Challenges of studying and processing dialects in social media. In *Workshop on Noisy User-generated Text (W-NUT)*.

Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. page 78.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. pages 1188–1196.

Wendy Liu and Derek Ruths. 2013. What's in a name? using first names as features for gender inference in twitter. In *Analyzing Microtext: 2013 AAAI Spring Symposium*.

Veronica Lynn, Youngseo Son, Vivek Kulkarni, Niranjan Balasubramanian, and H Andrew Schwartz. 2017. Human centered nlp with user-factor adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 1146–1155.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Dong Nguyen, Noah A Smith, and Carolyn P Rosé. 2011. Author age prediction from text using linear regression. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*. Association for Computational Linguistics, pages 115–123.

Barbara Plank and Dirk Hovy. 2015. Personality traits on twitterorhow to get 1,500 personality tests in a week. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. pages 92–98.

Daniel Preoţiuc-Pietro, Vasileios Lampos, and Nikolaos Aletras. 2015a. An analysis of the user occupational class through twitter content. In *ACL*.

Daniel Preoţiuc-Pietro, Svitlana Volkova, Vasileios Lampos, Yoram Bachrach, and Nikolaos Aletras. 2015b. Studying user income through language, behaviour and affect in social media. *PloS one* 10(9):e0138717.

Sravana Reddy and Kevin Knight. 2016. Obfuscating gender in social media writing. In *Proceedings of the First Workshop on NLP and Computational Social Science*. pages 17–26.

Sara Rosenthal and Kathleen McKeown. 2011. Age prediction in blogs: A study of style, content, and online behavior in pre-and post-social media generations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pages 763–772.

Bahar Salehi, Dirk Hovy, Eduard Hovy, and Anders Søgaard. 2017. Huntsville, hospitals, and hockey teams: Names can reveal your location. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*. pages 116–121.

Amos Tversky and Daniel Kahneman. 1973. Availability: A heuristic for judging frequency and probability. *Cognitive psychology* 5(2):207–232.

Svitlana Volkova, Yoram Bachrach, Michael Armstrong, and Vijay Sharma. 2015. Inferring latent user properties from texts published in social media (demo). In *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence (AAAI)*. Austin, TX.

Svitlana Volkova, Glen Coppersmith, and Benjamin Van Durme. 2014. Inferring user political preferences from streaming communications. In *Proceedings of the 52nd annual meeting of the ACL*. pages 186–196.

Svitlana Volkova, Theresa Wilson, and David Yarowsky. 2013. Exploring demographic language variations to improve multilingual sentiment analysis in social media. In *Proceedings of EMNLP*. pages 1815–1827.

Yi Yang and Jacob Eisenstein. 2017. Overcoming language variation in sentiment analysis with social attention. *Transactions of the Association for Computational Linguistics* 8:295–307.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 2979–2989.