Abstract

Due to its semantic succinctness and novelty of expression, poetry is a great test bed for semantic change analysis. However, so far there is a scarcity of large diachronic corpora. Here, we provide a large corpus of German poetry which consists of about 75k poems with more than 11 million tokens, with poems ranging from the 16th to early 20th century. We then track semantic change in this corpus by investigating the rise of tropes (‘love is magic’) over time and detecting change points of meaning, which we find to occur particularly within the German Romantic period. Additionally, through self-similarity, we reconstrukt literary periods and find evidence that the law of linear semantic change also applies to poetry.

1 Introduction

Following in the footsteps of traditional poetry analysis, Natural Language Understanding (NLU) research has largely explored stylistic variation (Kaplan and Blei, 2007; Kao and Jurafsky, 2015), (over time) (Voigt and Jurafsky, 2013), with a focus on sound devices (McCurdy et al., 2015; Hench, 2017) and broadly canonicalised form features such as meter (Greene et al., 2010; Agirrezabal et al., 2016; Estes and Hench, 2016) and rhyme (Reddy and Knight, 2011; Haider and Kuhn, 2018), as well as enjambment (Ruiz et al., 2017) and noun+noun metaphor (Kesarwani et al., 2017).

However, poetry also lends itself well to semantic change analysis, as linguistic invention (Underwood and Sellers, 2012; Herbelot, 2014) and succinctness (Roberts, 2000) are at the core of poetic production. Poetic language is generally very dense, where concepts / ideas cannot be easily paraphrased. With a distributional semantics model, Herbelot (2014) finds that the coherence of poetry significantly differs from Wikipedia and random text, allowing the conclusion that poetry is – compared to ordinary language – unusual in its word choice, but still generally regarded comprehensible language. Recently, there has been research with topic models on poetry with Latent Dirichlet Allocation. Navarro-Colorado (2018) explores the overarching topical motifs in a corpus of Spanish sonnetts, while Haider (2019) sketches the evolution of topics over time in a German poetry corpus, identifying salient topics for certain literature periods and applying these for downstream learning how to date a poem.

Following in this vein, we offer a method to explore poetic tropes, i.e. word pairs such as ‘love (is) magic’ that gain association strength (cosine similarity) over time, finding that most are gaining traction in the Romantic period. Further, we track the self-similarity of words, both with a change point analysis and by evaluating ‘total self-similarity’ of words over time. The former helps us to reconstruct literary periods, while the latter provides us with further evidence for the law of linearity of semantic change (Eger and Mehler, 2016) using our new method.

We do this with a model that learns diachronic word2vec embeddings jointly over all our time slots (Bamman et al., 2014), avoiding the need to compute the cosine similarity of two word vector representations on second order to align the embeddings.

Our contributions are: we (1) provide a large corpus of German poetry which consists of about 75k poems, ranging from the 16th to early 20th century with more than 11 million tokens.\(^1\) We then track semantic change in this corpus with (2) two self-similarity experiments and finally (3) by investigating the rise of tropes (e.g. ‘love is magic’) over time.

\(^1\)http://github.com/thomasnikolaushaider/DLK
2 Related Work

Semantic change has been explored in various works in recent years. One focus has been on studying laws of semantic change. Xu and Kemp (2015) explore two earlier proposed laws quantitatively: the law of differentiation (near-synonyms tend to differentiate over time) and the law of parallel change (related words have analogous meaning changes), finding that the latter applies more broadly. Hamilton et al. (2016) find that frequent words have a lower chance of undergoing semantic change and more polysemous words are more likely to change semantically. Eger and Mehler (2016) find that semantic change is linear in two senses: semantic self-similarity of words tends to decrease linearly in time and word vectors at time $t$ can be written as linear combinations of words vectors at time $t-1$, which allows to forecast meaning change. Regarding methods, Xu and Kemp (2015) work with simple distributional count vectors, while Hamilton et al. (2016) and Eger and Mehler (2016) use low-dimensional dense vector representations. Both works use different approaches to map independently induced word vectors (across time) in a common space: Hamilton et al. (2016) learn to align word vectors using a projection matrix while Eger and Mehler (2016) induce second-order embeddings by computing the similarity of words, in each time slot, to a reference vocabulary. Kutuzov et al. (2018) survey and compare models of semantic change based on diachronic word embeddings. Dubossarsky et al. (2017) caution against confounds in semantic change models.

An interesting approach besides computing independent word embeddings in each time period has been outlined by Bamman et al. (2014) who jointly compute embeddings across different linguistic variables: each word $w$ has an embedding

$$w = e_w W_{\text{main}} + e_w W_C,$$

where $W_{\text{main}} \in \mathbb{R}^{W \times d}$ is a main embedding matrix and $W_C \in \mathbb{R}^{W \times d}$ is an embedding matrix for linguistic variable $C$, and $e_w$ is a 1-hot vector (index) of word $w$. In their original work, $C$ ranges over geographic locations (US states). A joint model has several advantages: it better addresses data sparsity (for specific variables) and it directly learns to map words in a joint vector space without necessity of ex-post projection. In our work, we use this latter model for temporal embeddings in that each linguistic variable $C$ corresponds to a time epoch $t$:

$$w(t) = e_w W_{\text{main}} + e_w W_t$$

This dispenses the need to align independently trained embeddings for every time slot. Instead, a joint (MAIN) model is learned that is then re-weighted for every time epoch. While this is convenient, it does not necessarily mean that embeddings of a certain low-frequency word in a given time slot are stable. If there is not enough context for a given word in a certain time period $t$, the model just learns the MAIN embedding with little to no re-weighting, i.e., the matrix $W_t$ may not be well estimated (at certain rows).

Corpus

We compile the largest corpus of poetry to date, the German Poetry Corpus v1, or Deutsches Lyrik Korpus version 1, DLK for short. See table 1 for a size overview. We know of no larger poetry collections in any language. Only the collection from the English Project Gutenberg offers a similar size, but due to a lawsuit, as of 2018 it is not available in Germany anymore.\(^2\)

<table>
<thead>
<tr>
<th>Tokens</th>
<th>11,849,112</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines</td>
<td>1,784,613</td>
</tr>
<tr>
<td>Stanzas</td>
<td>280,234</td>
</tr>
<tr>
<td>Poems</td>
<td>74,155</td>
</tr>
<tr>
<td>Authors</td>
<td>269</td>
</tr>
</tbody>
</table>

Table 1: Corpus Size, Deutsches Lyrik Korpus v1

DLK covers the full range of the New High German language (of public domain literature), ranging from 1575 AD (Barock period) up to 1936 AD (Modern period). It is collected from three resources: (1) Textgrid\(^3\) (TGRID), (2) The German Text Archive\(^4\) (DTA), and (3) Antikoerperchen (ANTI-K). The latter two were first described by Haider and Kuhn (2018). All three corpora are set in TEI P5 XML.

TGRID offers around 51k poems with the label ‘verse’ (TGRID-V). Many of these texts have a unique timestamp. Where this is not the case, we take the average year between the author’s birth and death.

\(^2\)http://block.pglaf.org/germany.shtml
\(^3\)textgrid.de
\(^4\)deutschestextarchiv.de
DTA offers around 28k poems with the label ‘lyrik’ (DTA-L). The poetry in DTA is organized by editions (whole books), rather than by single poems. The timestamps are therefore guided by these few books, but give very accurate stamps.

ANTI-K is a collection of only 156 poems of school canon that was mined from antikoerperchen.de/lyrik. It has very accurate annotation, including literary periods, that allow us to gauge the distribution of poems according to canonic periods.

For training our model, we organize the corpus by stanzas, where every stanza represents a document. The reasoning behind this is that for poetic tropes, words are likely to stand in local context. We merge our collections and remove duplicate stanzas that match on their first line. This removes 9600 duplicates. Filtering Dutch and French material further eliminates 3200 stanzas. Since the earliest time slot 1575–1625 is too small, we merge it with the adjacent slot.

Figure 1: Distribution of stanzas in 50 year slots, 1575–1925 AD. Period labels: Barock (baroque), Aufklärung (enlightenment), Empfindsamkeit (sentimentalism), Klassik (Weimar classicism), Frühromantik (early romantic), Spätromantik (late romantic), Moderne (modernity). First slot (1600) is merged into the adjacent slot.

See figure 1 for the distribution of stanzas in 50 year time slots. The slots are labelled with approximate literature period information based on the clustered annotation in ANTI-K. We can see that the Romantic period (approx. 1750–1875) is overly heavy, while the Barock period is somewhat underrepresented.

We lemmatize based on a gold token:lemma mapping that was extracted from DTA-L in tcf format. Where this does not cover a token, we pos-tag the line with pattern.de to feed into germalemma. We publish our corpus in json format.

**Experiments**

**Self-similarity**

We investigate semantic self-similarity of words over time in two ways: (1) How does poetic dictionary change over successive time steps (change point detection), and (2) how does contextual word meaning change in total over the whole time frame with respect to the word’s frequency (laws of conformity and linearity)? We use a model with a 25+50 sliding window, where time steps increase by 25 years, with a window size of 50 years. This doubles the data and allows a more fine grained analysis.

**Pairwise Self-Similarity**

We compute how the contextual use of words changes over successive time steps. We do this by determining the self-similarity of a word \( w \) over time by calculating the cosine similarity of the embedding vectors \( w(t) \) for \( w \) at time periods \( t = t_i \) and \( t = t_{i+1} \) as in equation (1):

\[
\text{cossim}(w(t_i), w(t_{i+1}))
\]

where \( \text{cossim}(a, b) \) is defined as \( a^\top b \) for two normalized vectors \( a \) and \( b \).

Thus, we can aggregate the self-similarity for the most frequent words at every time step and plot the change for all these words combined. See figure 2 for a boxplot of this pairwise self-similarity for the 3000 most frequent words.

**Results**

Our interpretation is that rising similarity signifies a homogenization of overall word use (diction), while a falling similarity signifies semantic diversification. In particular, we see a steady falling trajectory in the period between 1600 and 1675, with a dip at 1700. This period is generally regarded as the ‘Barock’ period. Then, word use slowly homogenizes, until we see a sharp dip around 1750, the onset of the Romantic period. Then it homogenizes during the Romantic period, until a dip at 1850, the end of the Romantic period, and then a homogenization into the the onset of modernity.

---

5 https://github.com/WZBSocialScienceCenter/germalemma
6 http://github.com/thomasnikolaushaider/DLK
Figure 2: Pairwise Self-Similarity. Top-3000 most frequent words. Cosine similarities of word $w$ with itself in adjacent time slots $\text{cossim}(w(t_i), w(t_{i+1}))$

Total Self-Similarity

We determine change of word meaning across any possible time distances as a probing for the linearity of semantic change in our corpus.

For this, we calculate the semantic self-similarity of a word across all time periods $t_i$ and $t_j$ with $t_i < t_j$. We then aggregate all pairwise distances in years

$$\text{dist}(t_i, t_j) = |t_i - t_j|$$

for all words $w$.

To obtain robust estimates of embeddings, we only allow words that occur at least 50 times in every time slot and remove stopwords, leaving us with 472 words.

The x-axis in Figure 3 gives the distances $\text{dist}(t_i, t_j)$ while the y-axis shows the distribution of cossims over all words $w$ within each distance.

We find that there is approximately a linear relation between the distance of timeslots for an average word, where close slots are more similar, and far apart slots are increasingly dissimilar. However, the variance also increases with distance.

Additionally, we equally divide our words into a low-frequency and a high-frequency band. We find that the low-frequency band shows a generally higher self-similarity than the high-frequency band over all distances. This would mean that, overall, high frequency words tend to be more semantically diverse over time, i.e. stand in more diverse contexts. In contrast, low-frequency words stand in fewer contexts, therefore undergo less change. However, this could also come from the tendency of the model to revert to MAIN.

Emerging Tropes: Collocations & Metaphors

Method

To detect emerging tropes, we calculate the cosine similarity of word pairs over time. For the sake of visualization we use a 50+50 model with 6 time slots. We then perform Principal Component Analysis (PCA) over the resulting trajectories (Eger, 2010). The resulting principal components show that similar trajectories are co-variant. Component 1 aggregates stable high/low trajectories, while component 2 aggregates rising/falling trajectories. We illustrate our finding with the tropes for the concept ‘love’ (‘Liebe’ in German) and determine the most salient word pairs over the whole dataset. ‘Love’ is a very frequent word in poetry. Nevertheless, this approach works equally well for any word, except very low frequency words that show idiosyncratic behavior as they are not well distributed.

Results

We calculate the distance of ‘love’ against every other word $w$, where $w$ has to occur at least 30 times in the corpus, and it needs to be represented in every time slot at least twice. We allow one slot to be empty.

The first 4 components of PCA explain >.95 variance, where component 1 explains 73%, component 2 13%, and component 3 5%. We retrieve the top-25 word pairs at every component extreme.
<table>
<thead>
<tr>
<th>rising traj.</th>
<th>falling traj.</th>
<th>high traj.</th>
<th>low traj.</th>
</tr>
</thead>
<tbody>
<tr>
<td>frische</td>
<td>aufrechen</td>
<td>liebe</td>
<td>brummen</td>
</tr>
<tr>
<td>veilchen</td>
<td>alsbald</td>
<td>freundschaft</td>
<td>krähen</td>
</tr>
<tr>
<td>niedersinken</td>
<td>billigkeit</td>
<td>lust</td>
<td>rasseln</td>
</tr>
<tr>
<td>duftig</td>
<td>erzeigen</td>
<td>treue</td>
<td>rum</td>
</tr>
<tr>
<td>jenseits</td>
<td>unterstehen</td>
<td>trieb</td>
<td>bock</td>
</tr>
<tr>
<td>zauber</td>
<td>betragen</td>
<td>seligkeit</td>
<td>dum</td>
</tr>
<tr>
<td>entgleiten</td>
<td>stracks</td>
<td>hoffnung</td>
<td>prasseln</td>
</tr>
<tr>
<td>kinden</td>
<td>zuerkennen</td>
<td>glaube</td>
<td>trommel</td>
</tr>
<tr>
<td>hoffend</td>
<td>hierin</td>
<td>keusch</td>
<td>säbel</td>
</tr>
<tr>
<td>efeu</td>
<td>schmessen</td>
<td>treu</td>
<td>traben</td>
</tr>
<tr>
<td>enthüllen</td>
<td>anläß</td>
<td>erkalten</td>
<td>bellen</td>
</tr>
<tr>
<td>erfüllung</td>
<td>jederzeit</td>
<td>wahr</td>
<td>block</td>
</tr>
<tr>
<td>heimat</td>
<td>muben</td>
<td>immerdar</td>
<td>bügel</td>
</tr>
<tr>
<td>trübe</td>
<td>schimpfen</td>
<td>regung</td>
<td>gaul</td>
</tr>
<tr>
<td>gloria</td>
<td>stecken</td>
<td>gegenliebe</td>
<td>grasen</td>
</tr>
</tbody>
</table>

Table 2: Top 15 words per dimension for 'love' tropes from PCA extremes, plotted in figures 4, 5, 6 and 7.

We find that component 1 orders trajectories based on high/low semantic similarity, while component 2 orders based on rising/falling trajectories. See figures 4 (high trajectory), 5 (rising trajectory), 6 (low trajectory) and 7 (falling trajectory). See table 2 for the respective word pairs (collocations) with ‘love’ as they are plotted.

**Stable High Trajectories** Trajectories in figure 4 (table 2 column 3) have a consistently high cosine, meaning that these collocations have remained unchanged since the Baroque period: ‘love is fidelity’, ‘love is friendship’, or ‘love is lust’. These are conventional near-synonyms, just as (‘apple’, ‘tree’) or idioms (‘apples’, ‘pears’). A k-nearest neighbor (KNN) analysis retrieves these collocations. Performing this analysis for multiple words, we find that the idiom (‘apple’, ‘pear’) is a special case, as it strongly loads into both rising and stable high PCA dimensions (both top 20).

**Rising Trajectories** Figure 5 (table 2 column 1) shows rising collocations that emerge during the Romantic period, i.e. ‘fresh love’, ‘love is magic / enchantment’ and ‘love is violets’. A metaphorical (trope) interpretation is most likely here.

**Falling Trajectories** As illustrated in figure 7 (column 2), these collocations fall into obscurity.

---

8(‘Treue’, ‘Liebe’) 
9(‘Freundschaft’, ‘Liebe’) 
10(‘Apfel’, ‘Baum’) 
11(‘Apfel’, ‘Birnen’), ‘compare apples and oranges’. 
12(‘Frische’, ‘Liebe’) 
13(‘Zauber’, ‘Liebe’) 
14(‘Veilchen’, ‘Liebe’) 

---

15‘billigkeit (which changed from ‘equity’ to ‘cheap’) 
16aufrechen 
17betragen 
18trommel
References


David M Kaplan and David M Blei. 2007. A computational approach to style in american poetry. In Seventh IEEE International Conference on Data Mining (ICDM 2007), pages 553–558. IEEE.


