Beck vs. the Blues — an exercise in digital humanities

Introduction

What follows is an account of what I did for the assignment given in the *Introduction to Methods in Digital Humanities* -course. The task of the assignment was to take some dataset and process it in some way to yield an interesting analysis or visualisation.

Since, during the course, I realised I had forgotten practically everything (which wasn't much to begin with) I had previously learned about programming, I thought I'd better not get too ambitious and at least stick to what familiar with, which is corpus linguistics.

Way back, when I was taking a class in natural language processing, I was thinking it would be great to collect the lyrics of one of my all-time favourite songwriters, Beck, and apply some of the "magic" of NLP to maybe get a different sense of his work as a whole.

I was reminded of this when leafing through a list of the previous projects done for this course, and seeing an analysis of Hungarian love songs[1].

I collected all the lyrics from Beck's main 12 album releases (ones that are generally available) from a website, whiskeyclone.net [2] which I consider being the greatest and most authoritative repositories of Beck-related information. I reached out to the creator(s) of the webpage for a permission to use their data, but did not get a reply — therefore I will upload just the results.

Beck is known for his sometimes surreal, nonsensical and creative lyrical style. However, he has done many quite straightforward blues, folk and country inspired albums with fairly simple, bare-bones lyrics to match. He has often spoken about how very early on, he was influenced by blues in particular. Digging around the internet I came across a corpus of blues lyrics[3], so I thought why not compare the two.

And, as I was not feeling very confident about my programming skills, I thought I'd take up a fairly simple task and get the most frequently used words from each corpus. However, to get perhaps a more informative view, I decided to get a list of nouns, with commonly used words removed.

What I wanted to see was whether any of the more individual lyrical choices of Beck were still to be found in the most frequent word list, or whether they would be stripped away and reveal a fairly standard repertoire of pop-lyrical content. I suspected the latter, but was still hopeful that a sense of his themes would shine through. Also, I thought a comparison between the two corpora would be interesting.

Data

Beck

For the purposes of this project, I accepted the website creator's transcriptions and did not go back and check whether I agree with their judgement. Another important thing to point out is that *Whisceyclone* lists 1355 songs (some of them instrumentals, some of them covers), of which I collected just 170 (23 000 words after preprocessing, according to Pages).

The albums included are:

Stereopathetic Soulmanure

Mellow Gold
One Foot in the Grave
Odelay
Mutations (I included the bonus track *Runners Dial Zero*, since it is included in the copy I have on my shelf)
Midnite Vultures
Sea Change
Guero
The Information
Modern Guilt
Morning Phase

Blues

Michael Taft describes his corpus in the *Blues Lyric Formula* as consisting of “perhaps one-fifth of all the blues songs commercially recorded before World War One”[4]. (229 725 words after preprocessing, according to Pages). For more detailed look one can delve into the corpus files themselves.

Procedure

Pre-processing

I copy-pasted all the lyrics from 12 beck albums into a single plain text file, made it all lower case, standardised spelling of some verbs (*comin’* to *coming*) and removed the website author’s comments which were in square brackets, simply using find and replace in TextEdit. Next, I wanted to get rid of repeated (excessive repetition could very easily skew the results) and empty lines (removing empty lines was, I guess, at this point just an aesthetic choice — as I didn’t plan to investigate individual albums or songs, I just put everything together, one line after another).

Since such an easy solution turned out to be available, I used a web-based tool from Text Mechanic[5] to do this for me.
In doing so, I noticed that some duplicates were still present, which was due to some lines having trailing white spaces. Luckily, the website also has an easy tool for removing those.

Then, I removed all punctuation with a piece of code copied from here:

http://stackoverflow.com/a/12453619

Extracting most frequent nouns

Next, I wanted to get rid of stock words and extract all the nouns from the file, for which I used NLTK[6] for Python[7], with the help from the following:

http://stackoverflow.com/questions/33587667/extracting-all-nouns-from-a-text-file-using-nltk

I used code from the first post in which the person asks whether his is the most efficient way of doing this, and subsequently couple of commenters agree that it is not so bad.

For stop word removal, I turned to the following tutorial:

https://pythonprogramming.net/stop-words-nltk-tutorial/

As the default POS-tagger in NLTK seemed not to be the most accurate one, I opted for using Stanford POS-tagger[8] instead, which I installed using the instructions from https://github.com/alvations/nltk_cli

and changed the code accordingly.

I also wished to get a list of all the resulting words where each word is on a separate line, so that I could copy then into Numbers easily. I used user JBernardo's solutions for this which I found here:

http://stackoverflow.com/questions/6167731/printing-list-elements-on-separated-lines-in-python

The final code is called *nouns_code.py*.

Editing the noun list

The Stanford POS-tagger also left a lot to be desired; it failed to correctly tag words with contracted forms (*don't, there's*, etc). NLTK's stop nouns list also failed to cut out pronouns with contracted verb forms such as *it's* and *you're*. In addition, I also wanted to get rid of all the interjections like *yeah*s and *ahhh*s still left on the list. I proceeded to clean up the word list in OpenRefine[9] and TextEdit, erasing all stop words still present, and also stripping off contracted forms. In doing so, I noticed that the list contained quite a few instances of *let's* , so I removed those also.

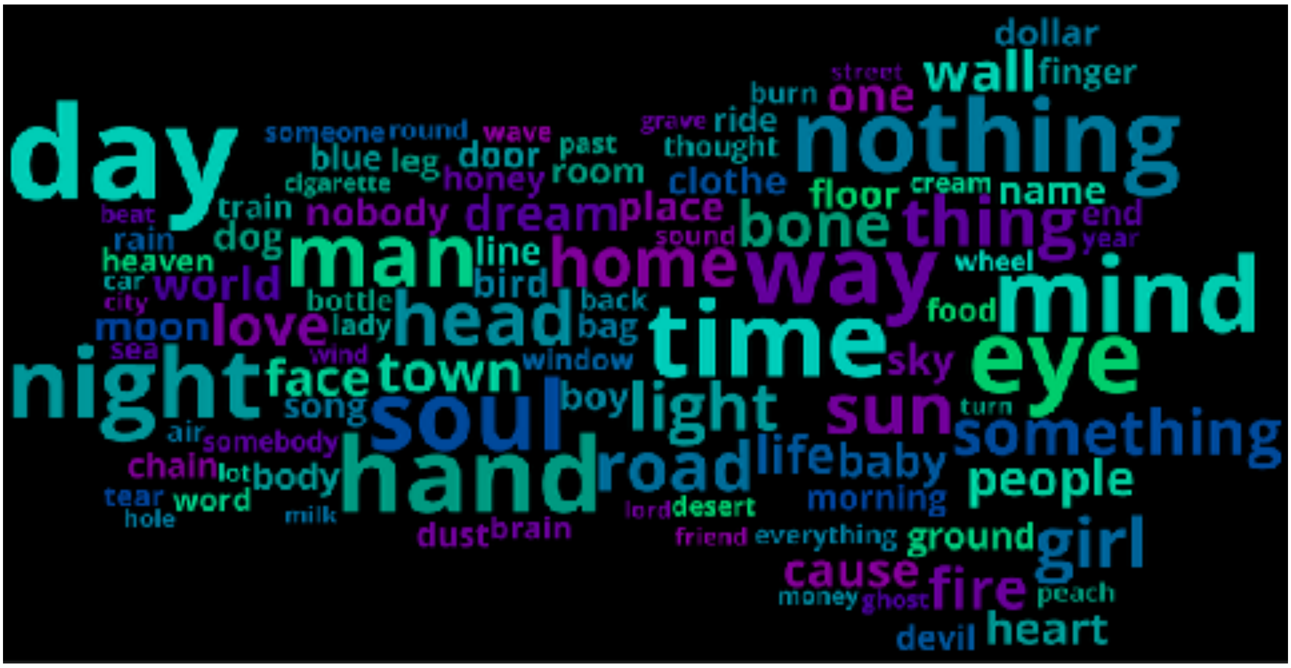
Then, I saved the results as a plain text file and got a list of the most common lemmas in AntConc[10] with the help of a version of the lemma list created by Yasumasa Someya, with no hyphenated words. (http://www.laurenceanthony.net/software/antconc/)

The results are in *beck_antconc_results.txt.*

I wanted to get approx. 100 top nouns, and all the nouns with frequency of 10 or more turned out to be in the top 102.

Next, for the visualisation part, I needed a tool that would let me input a weighted list of words, and Wordle[11] let me do just that. With Numbers and TextEdit I turned the top 102 nouns into a list where the word was followed by a colon and the frequency (*beck_nouns_top102.txt*).

The resulting word cloud image is *beck_nouns.png*.



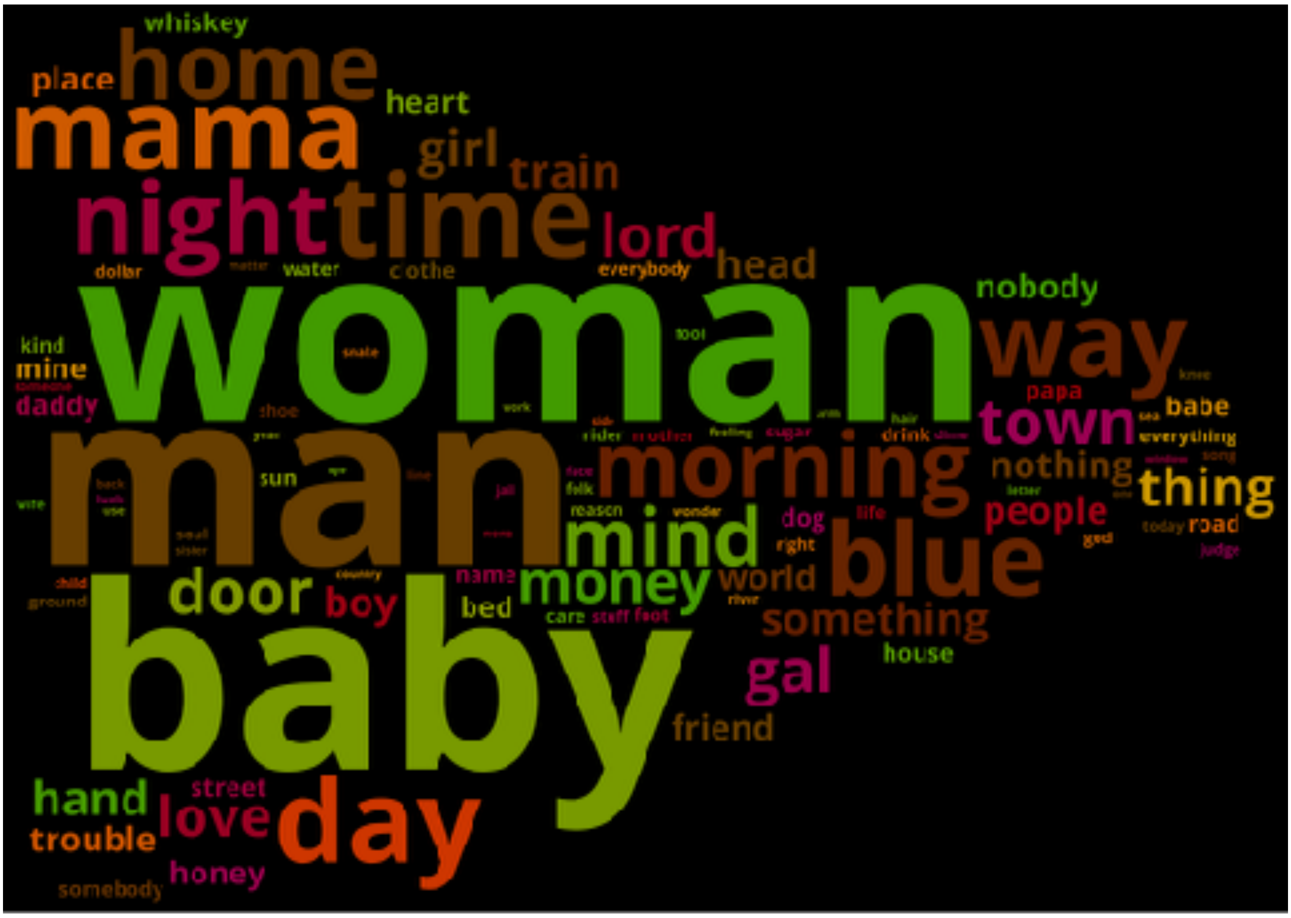
Then, I did the whole-she-bang to the blues corpus.

The corpus was in two files, which I merged into one. I separated the metadata (each line of metadata began with \C or \L, so I used Text Mechanic's tools again to erase all those lines, repeated lines, extra spaces, and empty lines.

Then, I followed the same procedure as with the Beck corpus, using TextEdit's find and replace and OpenRefine to clean up the list of nouns. The blues corpus had a fun genre-specific clean-up task with the *a*-prefix (*a-way, a-rolling, a-burning,* etc.).
The open refine project is attached and called *blues_nouns-csv.openrefine.tar.gz*..

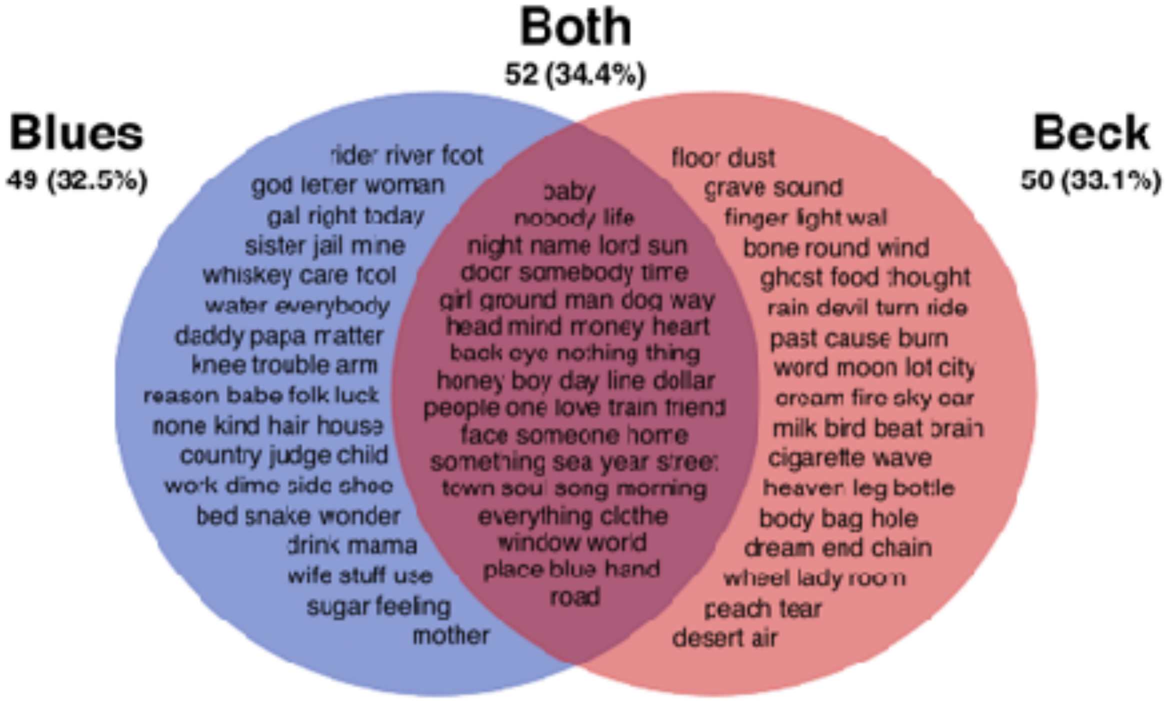
Investigating the AntConc results, I noticed *hey* (72/93) (and *ooo* (47,133) were still in the list, so I removed those as interjections were also removed from the Beck data.
The cut-off point was at 103, containing 101 words with frequency of 69 or more (*blues_nouns_top101.txt*).

Then, I made another word cloud with Wordle, *blues_nouns.png*.



Finally, I wanted to compare the two by making a Venn diagram, and for this, I used a tool provided by Bioinformatics and Evolutionary Genomics group at Gent university[12]

I could not find a tool that was free and would display the actual words inside the diagram, so I made one myself in Keynote using the image from the webtool, the resulting file is called *beck_vs_blues_venn.png*.



Results and discussion

This project was good practice in terms of (re)familiarising myself with python and NLTK and also using some of the tools introduced during the course, and web-based tools I discovered during this process.

I could have undoubtedly tried to automate more tasks with Python and NLTK, and the whole process surely could have been a lot more streamlined, but for some things I opted for solutions which enabled me to just move on with the task.

The credibility of the results do suffer from less than perfect POS-tagging. The final lists still contain entries which could have been omitted. For example, I was suspicious by the appearance of *use* (*use, uses* and *used*) on the list of blues nouns, as only the first two could possibly be used as nouns, and looking at concordances, its use as a noun seems to be in the minority. I also noticed a strange quirk (already present in the original corpus files) in the data as sometimes the word *because* had split in two (*beca, use*). There are other examples like this which could very well be contested on both of the lists.

The recall and precision of this noun-extraction method for these corpora is thus not known, but I did test whether I was getting more or less what I wanted with the code I scrambled together. Searching *night* in AntConc within the original Beck corpus file provided 41 hits, and the nouns file also 41. With *soul*, the results weren't as good, but good enough, with 36/33. Same figures for the blues files are (*blues_nopunct.txt/blues_nouns_all.txt*): *night*: 622/608; *soul* 99/99.

I could have also edited the lists further manually, but at least this way I made systematic and comparable changes to both lists, and did not make any more interpretative judgements.

However, the main question to be discussed is perhaps whether the methods used brought any additional value or insight into the lyrical works of Beck, or indeed the vocabulary of blues.

Looking at the words these two corpora have in common, they do seem to be quite standard fare. You have the women-folk, the source of so many conflicting feelings for (straight) male songwriters (*girl/gal, baby/babe, mama/mother, lady, wife,* and *sugar* with *baby and girl* appearing in both*),* there's *mind, soul, head* and *heart*, perhaps for exploring those innermost of human emotions. *Money* and *dollar* (also *honey*, which one often does not have unless one has money) are standard concerns. Familiar settings and transport feature in both: *place, street home, town, train, sea.*

In Beck's list, however, we find more words dealing with death or the afterlife (*grave, bone, ghost, heaven, devil, body)* and perhaps more abstract nouns and etherial and natural elements (*dust, light, wind, thought, rain, burn, fire, sky, dream, air, moon)* whereas traditional blues songs seem to be about more concrete issues (*trouble, matter, reason*) one faces in daily life (j*ail, judge, whiskey, bed, house, work*), and as the top three nouns are *baby, woman*, and *man*, we can perhaps assume that relationships are one of the most common cause for the blues.

It goes without saying that any serious attempt at analysing these results would require actually looking at the contexts in which these words appear. For example, what kind of *days* (a word which appears in both corpora's top five) are we talking about? *Test tube stillborn days?* (Beck, Novacane) Or *meatless and wheatless days*? (Blind Lemon Jefferson, Rabbit Foot Blues)

Still, a general knowledge of modern western music, starting with the blues, already gives you confirmation that these results largely conform to expectations of the kind of lyrical content we are all used to hearing. For a reference, here is a link to a word cloud of the most common words in the titles of songs that have been on the Billboard chart:

http://musicthing.blogspot.fi/2008/05/100-greatest-ever-cliches-in-pop-song.html

In short, the results conformed to expectations, but also brought out some interesting differences in lyrical content between the two corpora.

As for the Beck corpus — for it to be an actual proper corpus, it would preferably include every known song of his with metadata, for a start. For a body of work of a single person, still relatively small, I am not sure whether, for example, topic modelling would bring any additional insight. Still, a properly complied and organised corpus would open up a lot more possibilities for a fine-grained analysis.

[1] tiirikainen, . (2016). *Hungarian-love-songs: Hungarian love songs - First commit* [Data set]. Zenodo
http://doi.org/10.5281/zenodo.44570
Accessed 20.12.2016

[2] http://www.whiskeyclone.net/
Accessed 4.12.2016

[3] Taft, M. (1983) *Blues lyric poetry : an anthology* [Electronic resource]/. Oxford: University of Oxford.
Available from http://ota.ox.ac.uk/desc/1409.
Accessed 20.12.2016

[4] Taft, M. (2006) The Blues Lyric Formula. Routledge, New York. p.5
Available from Google Books: https://books.google.fi/books?id=M78Mn3MNfv4C&lpg=PA2&ots=iad92rguxz&dq=blues%20lyric%20formula&hl=fi&pg=PA5#v=onepage&q=blues%20lyric%20formula&f=false
Accessed 20.12.2016

[5] http://textmechanic.com
Accessed 18.12.2016

[6] Bird, Steven, Edward Loper and Ewan Klein (2009).
Natural Language Processing with Python. O'Reilly Media Inc.
Available from http://www.nltk.org/book_1ed/
Natural Language Tool Kit For Python
http://www.nltk.org.

[7] https://www.python.org Version 2.7

[8] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003.
Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network.
In Proceedings of HLT-NAACL 2003, pp. 252-259
Available from http://nlp.stanford.edu/software/tagger.shtml

[9] http://openrefine.org/index.html Version 2.6.-rc.2.

[10] Anthony, L. (2014). AntConc (Version 3.4.3) [Computer Software]. Tokyo, Japan: Waseda University.
Available from http://www.laurenceanthony.net

[11] http://www.wordle.net
Accessed 20.12.2016

[12] Calculate and draw custom Venn diagrams
http://bioinformatics.psb.ugent.be/webtools/Venn/