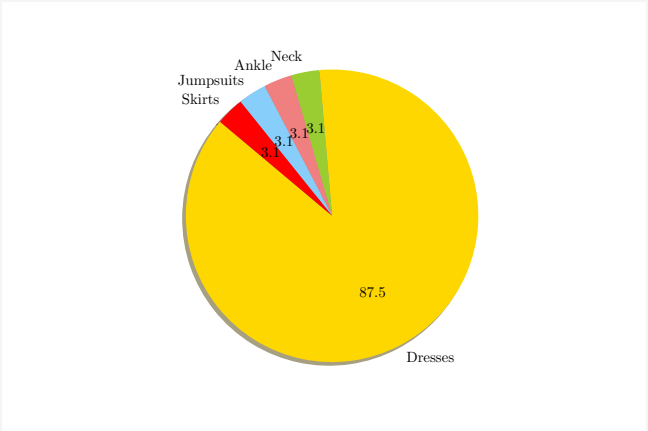
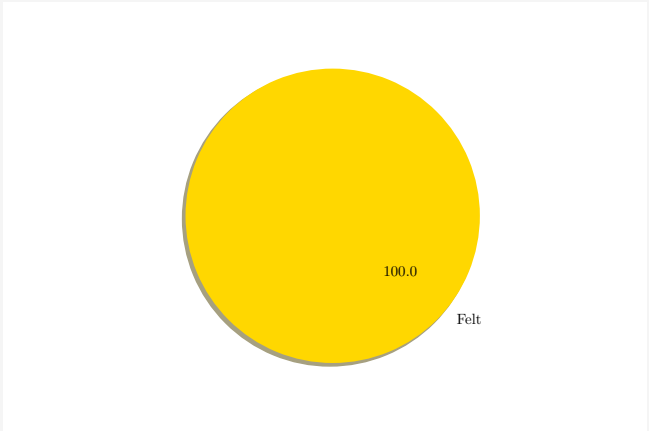
Outline

Data Exploration Review

Clothing item detection easier than material/brand



(a)



(b)

(c)

Figure: a), b) contains identified items/materials respectively for c). Detecting items based on text data looks promising. Detecting brand/material/pattern is harder... due to data sparcity & homonyms

Data Exploration Review

Shortcomings of Pre-trained vectors

First Word	Second Word	Vectors trained on Wikipedia ($\mathcal{V}_1$)	Vectors trained on Instagram posts ($\mathcal{V}_2$)
#gucci	gucci	0	0.85
gucci	prada	0.82	0.76
sweater	shirt	0.72	0.89
coat	jacket	0.68	0.94
jacket	top	0.35	0.92
blouse	top	0.31	0.81
👖	jeans	0	0.41
👖	dress	0	0.08
👗	dress	0	0.31
👜	handbag	0	0.62
👜	dress	0	0.11
👠	heels	0	0.48
👠	shoe	0	0.57
👟	sneaker	0	0.83

Table: Cosine similarity between fashion words/tokens.

Challenges

Noisy Data

We know the data is noisy, below I quantize *how noisy*.

Text Statistic	*Fraction of corpora size*	*Average/post*	*St. Dev*	*Min post*	*Max post*
Number of emojis	0.15	48.63	141.15	0	17938
Number of hashtags	0.03	9.14	12.48	0	1325
Number of user handles	0.06	18.62	232.74	0	46208
Number of OOV words	0.46	145.02	477.89	0	58832

Table: Measurements of lexical noise in the corpora under study. The vocabulary used to compute Out of Vocabulary (OOV) words is a vocabulary containing 3 million unique words. The vocabulary is provided by Google for research purposes, and is based on newswire text collected from a corpora of google news articles, consisting of 100 billion words.

Challenges

Data Sparsity

Mean length of captions is 29 words, mean length of comments is 6 words.

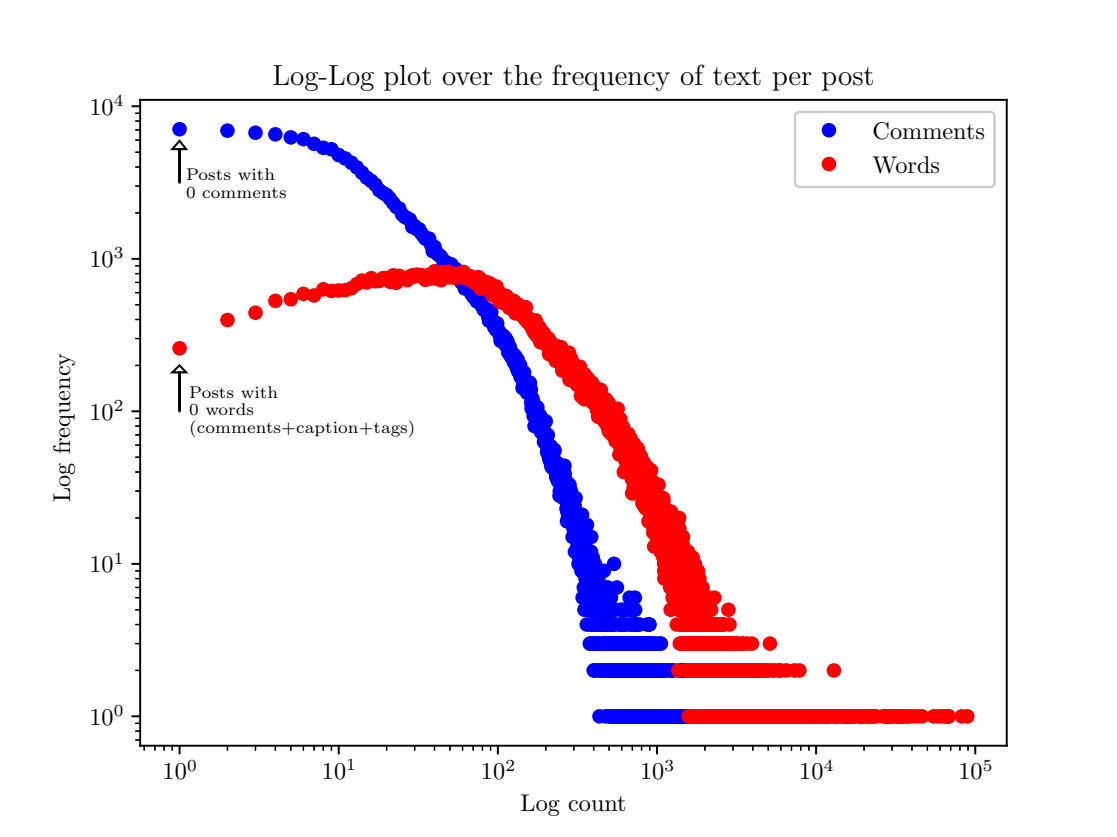


Figure: The distribution of text over Instagram posts in the corpora under study. Roughly power law, most posts contains between 0-100 words, but some posts contain over 100 000 words.

Challenges

Not only fashion topics

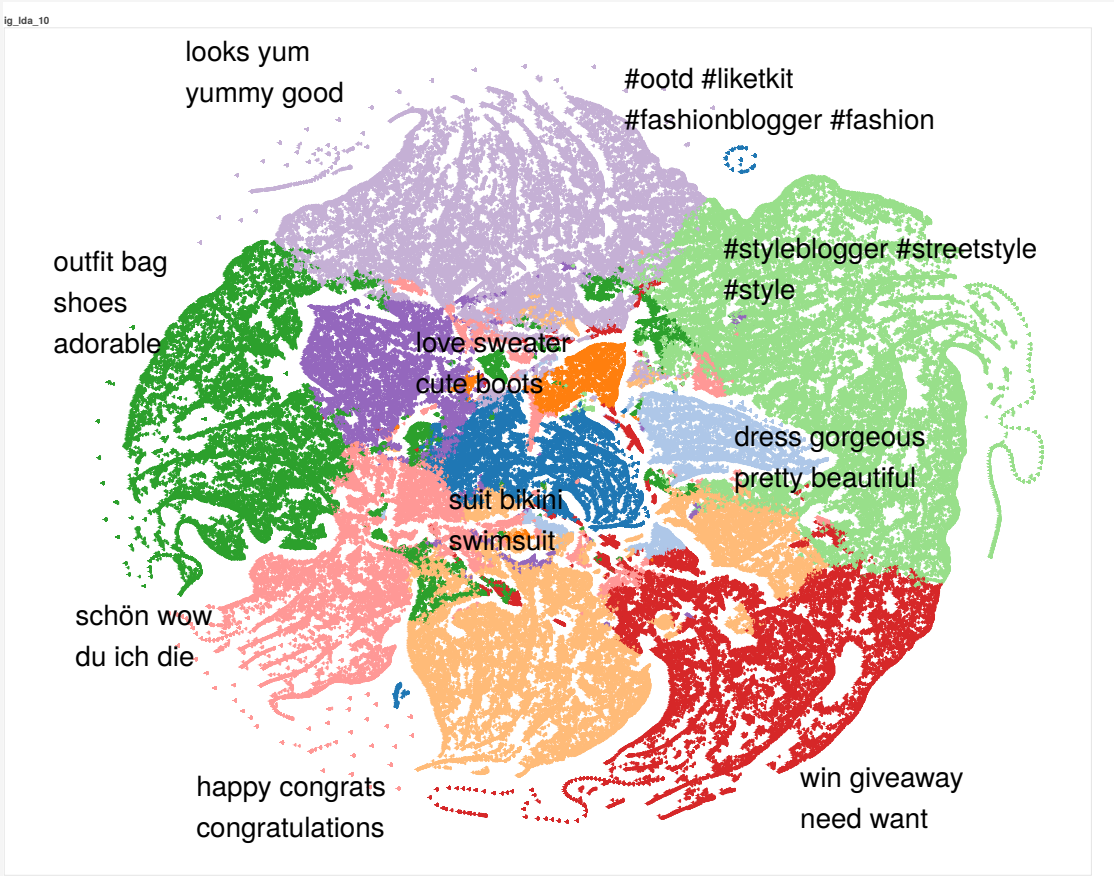


Figure: LDA-10 topics. 6/10 topics are fashion related, rest are: food, a set of german comments (also fashion related but less useful), giveaways, birthdays. **97 languages** identified among comments (high majority in English, second highest is Chinese on 6%, German on 3%).

Corpora Homogenity

1. **High corpora similarity** - as expected since all posts are from a single domain.
2. χ^2 score of 46 (lower score indicates more similar), can be compared to related work that obtained χ^2 score of 556 on twitter corpora not focused on any domain.
3. $\chi^2 = 46$ basically mean we cannot reject null hypothesis that the instagram posts in our corpora contains words draw from the same distribution (p-value $= 0.17$).

Text Normalization

Problem

Problem: Traditional NLP tools break down on social media text (cannot rely on PoS, capitalization, pre-trained word vectors, WordNet, vocabularies, single language etc).
Two main approaches in related work:

- Do regular text normalization and adapt NLP tools to the social media domain
- Do specialized text normalization to de-noise the text (spelling correction, exclude emojis, hashtags etc)

Text Normalization

Approach

From my impression, **adapting NLP tools to our domain** looks most promising. This entails:

- Training our own word vectors instead of using pre-trained.
- Keeping hashtags + emojis in the corpora
- Designing our own information extraction techniques that work in our domain
- Use other sources of supervision than Lexical resources such as WordNet

Word Vectors

Only Preliminary Results - Need to fine-tune evaluation to be more significant (need larger sample size (evaluation set) to reduce p-value of null-hypothesis)

Domain specific vectors out-perform pretrained vectors on domain-specific evaluation (worse in general evaluation).

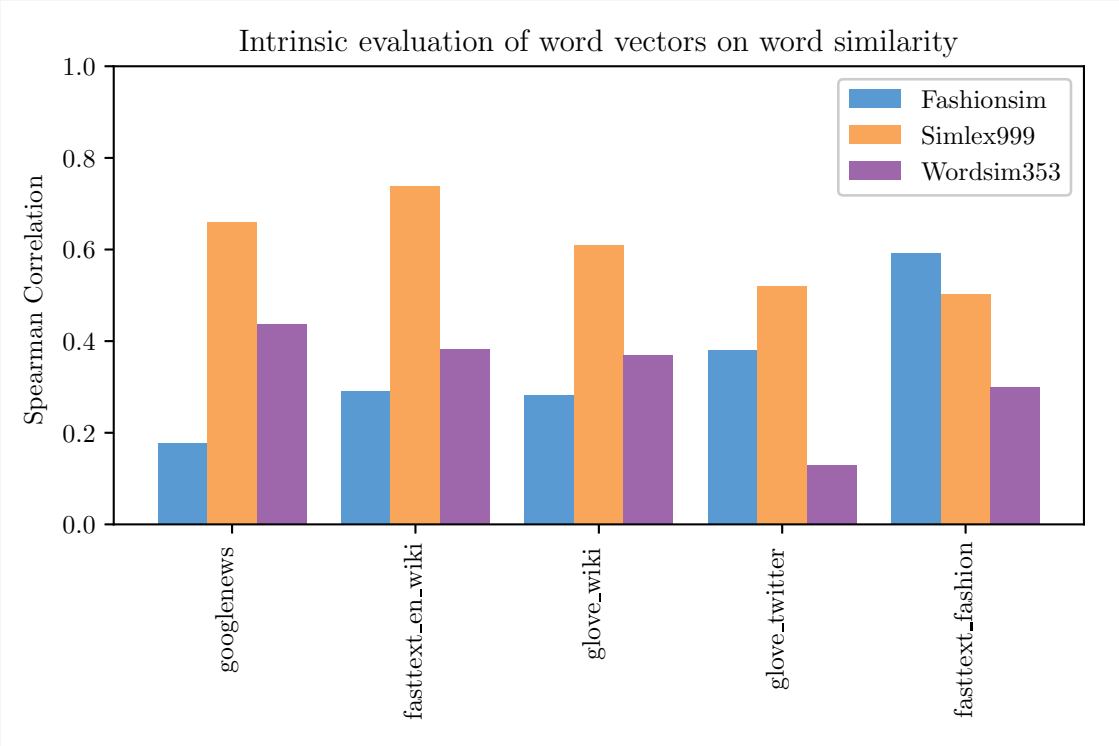
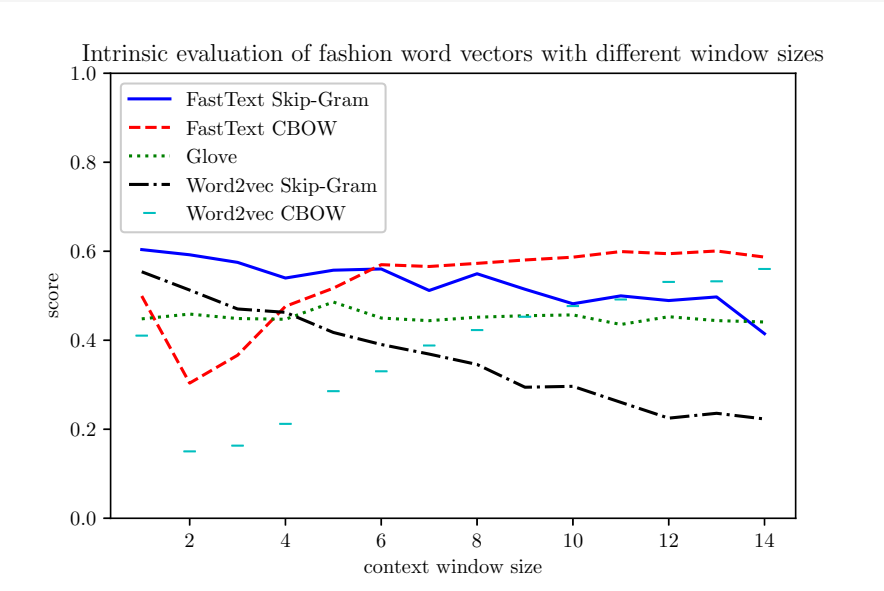


Figure: Comparison between pretrained vectors and vectors trained on the fashion corpora with the FastText algorithm.

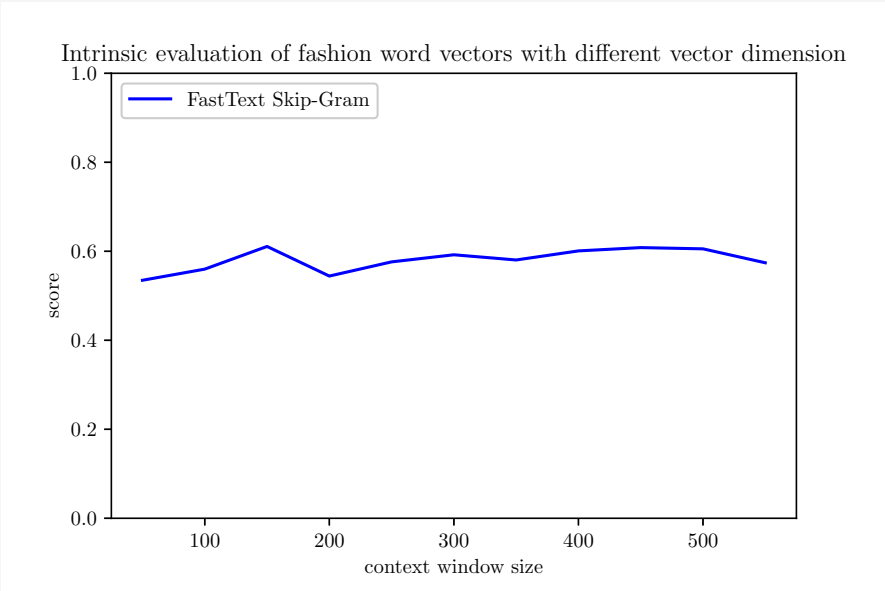
Word Vectors

Hyperparameters

We can make a contribution to find the best hyperparameters and algorithm for social media corpora.



(a) Different context window sizes.



(b) Different dimensions.

Figure: Evaluation of different algorithms on the fashion corpora with different context-window sizes and dimensions.

Unsupervised Information Extraction

Word Vector Clustering

Cluster words based on in which context they appear in the text $\Longrightarrow$ semantically similar words will be close. Classify Instagram post based on closest clusters in terms of item, brand, material etc.

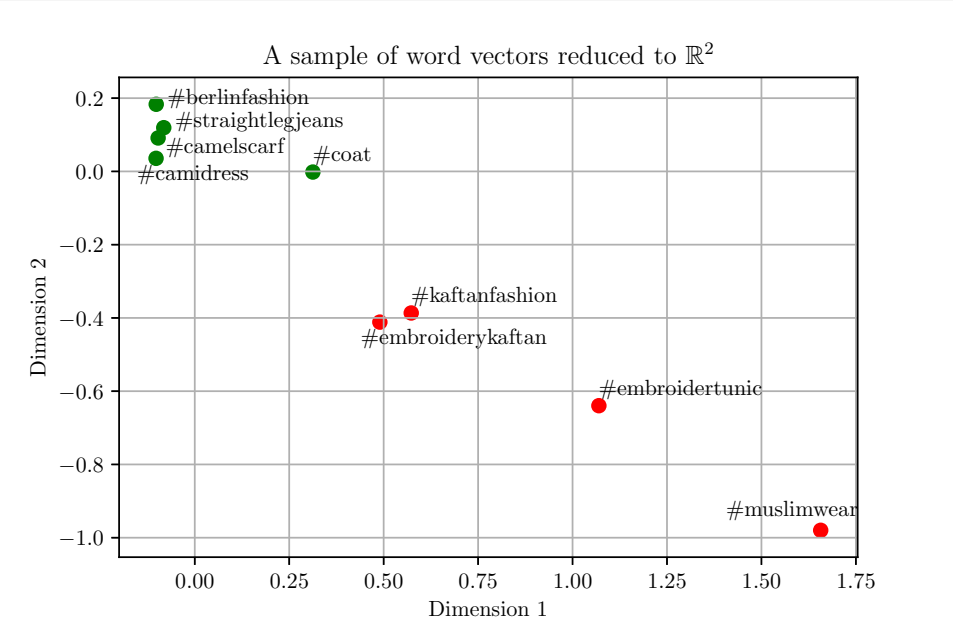


Figure: Two-dimensional PCA projection of the 300-dimensional skip-gram fasttext vectors trained on the fashion corpora. The plot includes words close to "#berlinfashion" and words close to "#kaftanfashion".

Unsupervised Information Extraction

External API

Problem 1: Certain brand and material names are Homonyms, such as brand called Hope, material called Felt.

Solution:

- Combine Semantic Similarity with Syntactic Similarity (Edit distance),
- Rank by frequency of occurrence in the text
- Lookup brands/materials in Semantic Web API's. Microsoft/Google has open APIs for this. The result of looking up "Hope" in probase yields:

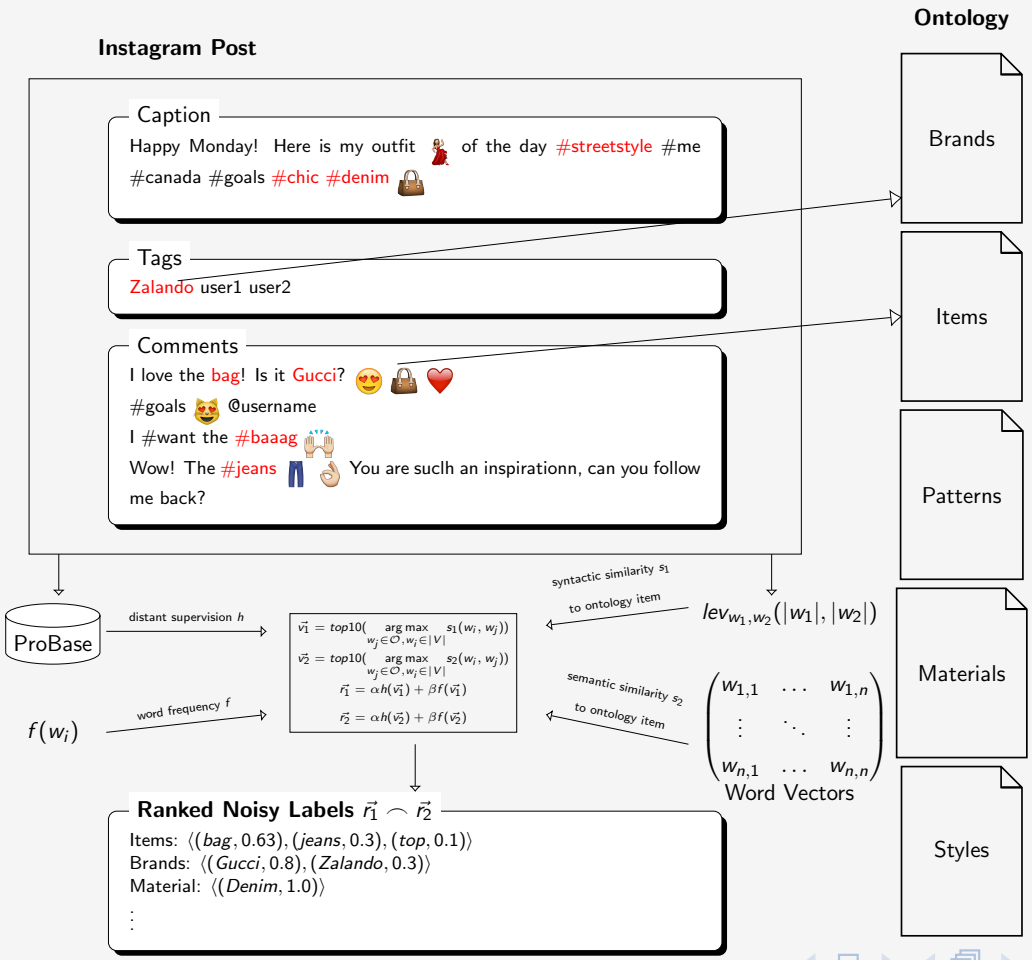
  ```
  {
  "emotion":0.210526315, "word":0.1862348178, ...
  }
  ```

  Very low score for "company" $\Longrightarrow$ Re-rank brands/materials based on this lookup result.

Unsupervised Information Extraction

Pipeline

Combine word frequency, semantic cluster, syntactic cluster, and distant supervision in linear combination to assign noisy labels.



Information Extraction & Classification

Limitations

Problem 2: In general, majority of posts do not include information about material and pattern. Brands occur a bit more often but still many post do not state the brand in the text either.

Solution: ? Have one class of "Unknown material/pattern/brand" in the classificiation. Hopefully Image classification can solve this.

Literature Study

Have studied so far:

- General NLP
- Text Normalization
- Word vectors
- General Machine learning and Machine Learning + NLP integration
- Semi-supervised learning
- Ontologies
- Social media text mining

Next:

- Latent Variable Models (Interesting results in related work)
- Concrete Classification Models (to apply on labelled data)
- More unsupervised learning techniques in case labelled data gets delayed

Future Work

- More experiments with Word Vectors
- Retrofit Word Vectors to achieve best possible accuracy in our domain
- Distributed Training of Word Vectors? FastText is only single machine $\implies$ bottleneck currently.
- Thesis writing, background + introduction + method draft by end of month
- Clean up the code
- Try to improve the unsupervised clustering classification
- Prepare for classification with labels from Amazon Turk