High-Precision Phrase-Based Document Classification on a Modern Scale

Ron Bekkerman, LinkedIn
Matan Gavish, Stanford
Leap into the future

- Who predicted in 1990 that all of us would have cell phones by 2000?
- Who predicted in 2000 that all of us would have GPS devices by 2010?
  - http://www.cs.technion.ac.il/~ronb/city.html
- Here's my prediction:
  - By 2020, we all will carry devices that record audio (video?) 24/7
  - And transmit the recorded data instantly to a cloud storage locker
- Why?
  - Personal security
  - “Perfect” memory
- Wanna analyze this data? Slice it? 😊
  - E.g., what are Libyans talking about today?
- The best approximation of “objectivity”
Simple arithmetic

- 1 billion people $\times$ 1 gigabyte of audio per day = 1 exabyte
  - 1 zettabyte in 3 years
- 1 billion people $\times$ 1K utterances a day $\times$ 100 bytes of text = 100 terabytes
  - Assuming that speech recognition works 😊

- Lexicon of an average person is ~10,000 words (or “concepts”)
  - Concepts are mostly overlapping in all languages
- 1 trillion utterances a day built from a 10K word set
  - How much repetition do we have?
The task

- Categorize 1 trillion short documents (utterances?) by topic
- Allow multi-labeling
- Allow reject option (“this utterance doesn’t belong to any category”)

- <10 words on average per utterance is not too much content
- Most of the signal sits in a few phrases (noun phrases?) per utterance
- Classify phrases and get utterance classification for free
  - (Semi-)manual labeling of phrases would be most accurate
  - English only (assuming machine translation works 😊)

- Example: “How many presents have you got to buy yet?”
  - Assign a phrase “buy presents” into the category “Shopping”
A $1M question: how many phrases to label?

- Our finding: not too many 😊
- Naively, the number of bigrams composed of 10K words is $10K^2 = 100M$
  - Forget about trigrams, fourgrams etc 😊
- But not all those phrases are common in the language!
  - Labeling low-frequency phrases is not cost effective
- **Number of common phrases composed of 10K words is under 1M**
  - Assuming 1¢ per phrase, you spend $10K (+fees) to label all of them
  - And two weeks of work 😊
Where did we take the data from?

- We took phrase counts from the Web1T data
  - Web1T has ngram counts from a one-trillion-word data collection
- Preprocessing:
  - Filtered out ngrams that appeared <1000 times in Web1T
  - Removed stopwords from all ngrams
  - Lower-cased all words
  - Ignored word order (by sorting words in an ngram)
- Example: “all Words from the Dictionary” → “dictionary words”
- Resulting dataset:  
  - 2.5M unigrams
  - 13M bigrams
  - 10M trigrams
  - 4M fourgrams
  - 1.4M fivegrams
Which phrases are considered common?

- Define frequency $F$ of a set $X$ as frequency of its least frequent item:
  $$F(X) = \min_{x \in X} F(x)$$
  
- Given a set of words $W$ and a set of phrases $T$ composed of words $W$ we say that $T$ is frequent enough if $F(T) \geq F(W)$
  - $T$ is as frequent as the set of words it was composed of

- We want to find an upper bound on the size $|T|$ as a function of $|W|$
Number of phrases: unrealistic setup

- Take $n$ most common phrases
- Identify $k$ words out of which those phrases were composed
- Take all $n^*$ common phrases composed out of those $k$ words
- Plot $n^*$ as a function of $k$

**Result:** $n^* = k^\alpha$ where $\alpha$ is small
- $1 \leq \alpha \leq 1.3$
- Number of phrases is under 0.5M
In classification by topic, we mostly care about topical words

- Compose a set of words $W$ that are topically related
  - Sampled from the most common 50K, 100K, 150K, or 200K words
- **Number of frequent enough phrases stays under 1M**
Web1T data is sufficient

- Sixgrams (and longer) won’t contribute too many common phrases
- The last graph is over the real-world controlled vocabulary
For text datasets of any size, there won’t be too many common phrases to label.
Phrase-based classification: formal requirements

- Classify **short** pieces of text (short documents)
  - Tweets, news headlines, Quora questions, helpdesk inquiries etc
- To a **small** number of categories
- The number of documents can be **huge** (billions? trillions?)
- Precision requirements are **high**: 95% and up
- Coverage should be **acceptable**
  - Some documents may remain uncategorized

Makes our life easier

Makes our life harder
‘Short document’ constraint relaxation

- Phrase-based classification may work **not only on short documents**
- Documents may be long but most of their topicality is in a few phrases

- Define $I(D;C)$ the Mutual Information between docs $D$ and their classes $C$
- Define $I(T;C)$ the Mutual Information between phrases $T$ and classes $C$

- Data Processing Inequality: $I(D;C) \geq I(T;C)$
- Phrases are **near sufficient** if $I(T;C) \geq (1-\varepsilon) I(D;C)$ for a small $\varepsilon$

- On our test set: $\frac{I(T;C)}{I(D;C)} = 0.92$ however $\frac{I(W;C)}{I(D;C)} = 0.53$
  - Much more information in phrases than in single words
How to classify phrases

1. Agree upon Taxonomy of Classes
2. Create Controlled Vocabulary
3. Build Common Phrases
4. Crowdsources Phrase Classification
5. Check Classification Consistency
6. Finalize Phrase Classification

Annotator Feedback
- Precision Goals not Met?
- Coverage Goals not Met?
Phrase classification step by step

- **Agree upon Taxonomy of Classes:**
  - Too bad if the chosen classes don’t represent the data $D$
- **Create Controlled Vocabulary:**
  - From a list of most common words in $D$, choose topical words $W$
- **Build common phrases:**
  - Out of words $W$, construct all phrases that appear in $D$
  - Choose the set $T$ of most common phrases
  - Filter out compound phrases (e.g. “machine learning and physics”)
  - Filter out too specific phrases (“content-based collaborative filtering”)
  - Filter out near redundant phrases (remove “recommendation systems”, leave “recommender systems”)

LinkedIn Talent Advantage
Crowdsourced Phrase Classification

- Educate the workers about the task
- **Multi-labeling**: Each phrase will be assigned to zero, one, or more classes
- Each phrase should be categorized by a number of workers
  - Use a voting mechanism to decide on a category
Check Classification Consistency

- Build a representation of each phrase \( t \), as an aggregation of all documents \( t \) belongs to
  - May add more information if available
- Using those representations, find \( k \) most similar phrases to each \( t \)
- Mark \( t \) as ‘suspicious’ if \( C(t) \neq C_{knn}(t) \)
- Over all non-suspicious phrases, learn a linear SVM
- For each suspicious \( t \), choose either \( C(t) = C_{svm}(t) \) or \( C_{knn}(t) = C_{svm}(t) \)
  - If \( C(t) \neq C_{knn}(t) \neq C_{svm}(t) \), leave \( t \) uncategorized
- Finalize phrase classification
  - Spot-check the resulting classification
Phrase-based classification is simple

- Given a document $d$ and a list of categorized phrases $T$
- Find all phrases from $T$ that appear in $d$: $T_d = d \cap T$
  - Use shallow NLP to filter out some irrelevant phrases
- Assign $d$ to all classes to which $T_d$ belongs: $C(d) = \bigcup_{t \in T_d} C(t)$
Deployed system: Job Title Classification

- About 100M job titles, a couple of dozen classes
- **Built a controlled vocabulary** of about 2500 words
  - Profession names, seniority words, and job function words
  - Mapped abbreviations, common misspellings, and foreign words onto the vocabulary words
- **Constructed most common phrases** (about 20K of them)
  - Removed compound, too specific, and redundant phrases
- **Crowdsourced phrase classifications**
  - Used two teams of LinkedIn employees
  - About 15% didn’t gain the majority vote, which got relabeled
  - About 25% titles were categorized inconsistently with the kNN model
    - Run SVM on them – resolved almost all issues
  - Final spot-checking was quite helpful too
Evaluation framework: comparison with SVM

- Once all documents got categorized, we got a 100M labeled collection
  - 95% precision, 80% coverage
- We choose 20K documents and trained an SVM on them
  - Tested over all 100M titles
- Tried 4 options for choosing the training set
  - (a) The categorized phrases $T$
  - (b) $T$ with word translations
  - (c) Most frequent titles
  - (d) Randomly chosen titles
- Reporting 2 quality measures:
  - Partial match in the assigned classes between our method and SVM
  - Full match in the assigned classes between our method and SVM
Results

<table>
<thead>
<tr>
<th>Setup</th>
<th>Partial Match</th>
<th></th>
<th></th>
<th></th>
<th>Full Match</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F-measure</td>
<td>Precision</td>
<td>Recall</td>
<td>F-measure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a)</td>
<td>91.2%</td>
<td>83.6%</td>
<td>87.2%</td>
<td>80.9%</td>
<td>65.5%</td>
<td>72.4%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b)</td>
<td>94.0%</td>
<td>92.1%</td>
<td>93.0%</td>
<td>82.4%</td>
<td>73.6%</td>
<td>77.8%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c)</td>
<td>95.4%</td>
<td>88.2%</td>
<td>91.7%</td>
<td>83.0%</td>
<td>73.4%</td>
<td>77.9%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(d)</td>
<td>95.1%</td>
<td>88.0%</td>
<td>91.4%</td>
<td>80.7%</td>
<td>73.7%</td>
<td>77.1%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 90 percentile means that phrase-based classification and SVM are not too different
  - Which looks true if we match only one class
  - It is not true if we match all classes in the multi-labeled setup
  - Humans are good at assigning documents to more than one class
Conclusion

- Proposed a general framework for classifying large document collections
  - If the “near-sufficiency” property folds (documents are short)
- The process is semi-manual with a heavy-lift consistency check
  - The consistency check boosts precision by ~20%

- Showed that regardless the size of the data, the semi-manual process is feasible
- Looking forward to categorizing trillions of utterances
  - See you all in 2020 😊