Context Is Everything: Finding Meaning Statistically in Semantic Spaces

Overview

- Simple, explicit measure of **contextual word importance**
- Supports **tiny contexts** (10+ sentences)
- Uses document word vector-cloud properties
- Contextually significant words define meaning
- Weighted **bag of words** model:
  - Substantially **outperforms state-of-the-art** for subjectivity analysis and paraphrase detection
  - Comparable to SotA for other **transfer learning** tests
- Applications:
  - A **better sentence vector baseline**
  - Easy sentence/document summarizer via **pathfinding**
  - Contextual stop word identification
  - Improved (and context-aware) cosine distance

Current Implementation Limitations

- Long Short-term Memory (LSTM) Networks:
  - Limited because **short-term**
  - Document-specific context ⇒ overfitting
- tf-idf sentence embedding (vector) baseline
  - **Rarer** words are **more important**
  - Essentially sum of tf-idf weighted word vectors
  - Requires **large** document, no handling of out-of-context words, **stratified** for rare words, ignores word **similarity**
- State-of-the-art global context approaches:
  - context vectors, deep structures, etc. (Black boxes)
  - Unsupervised **barely outperform tf-idf baseline**

Motivation

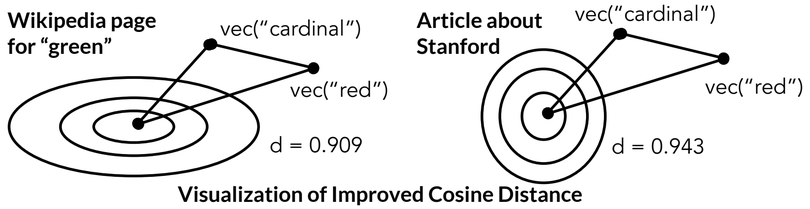
- Knowing what you're reading affects interpretation
- tf-idf baseline requires a large context dataset to work
- But people don't need a ton of text to establish context
  - Newspaper articles
  - Short stories
- Currently no simple baseline for global context

Datasets and Evaluation

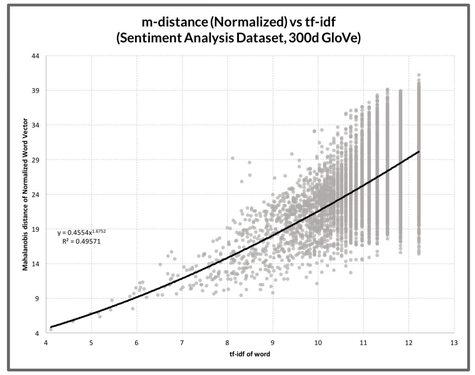
- Words / Clauses
  - Stanford Sentiment - Diverse context 9k examples
  - 300 dimensional pretrained GloVe (42b CC) - No out-of-vocabulary keys
- Sentences
  - SentEval train/dev set: Variety of transfer learning contexts
  - fastText vecs: 600b token CC, out-of-vocab support

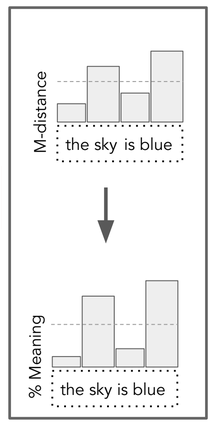
Algorithms: Word Vector Clouds

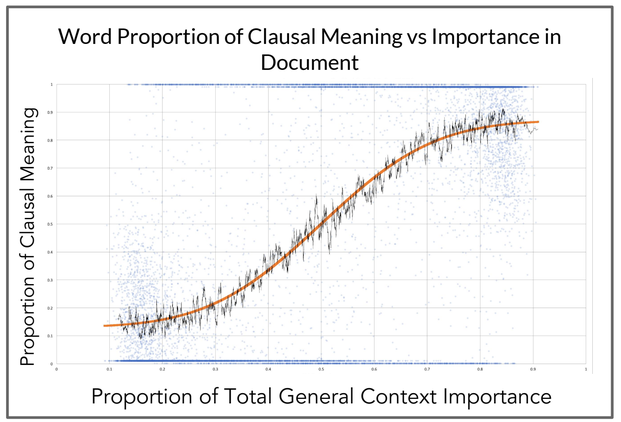
- Replacing tf-idf
  - Mahalanobis distance: Normalizes for stdev and covariance
  - Distance from document word-vector cloud
  - Needs only document word-vector covariance and average
  - Works with tiny data, since word vec dimensions are normal
- Better and Context-Aware Cosine Distance: $\cos C = \frac{a^2 + b^2 - c^2}{2ab}$



Visualization of Improved Cosine Distance







Test	Baseline	Unsupervised SOA (SOA*)	BOW
CR: Product Reviews Sentiment	79.2	**83.1** ST-LN	**79.42**
MPQA: Opinion Polarity	82.4	**89.3** ST-LN (90.2^{13})	**88.28**
SUBJ: Subjectivity/Objectivity	90.3	93.7 ST-LN (95.5 AdaSent)	**99.6**
TREC: Question Type (who, what, etc)	**85**	**92.2** ST	79.6
MRPC: Paraphrase Detection	73.6/81.7	73.0/82.0 ST	**74.72/82.16**
SICK-E: SICK Entailment	-	**84.6** SIF (86.3^{11})	**78.02**
SST: Movie Review	-	**82.9** ST (84.6^{11})	**81.38**

Algorithms: Sentence Embeddings

- Unified Clause-Word Vector Space
  - GloVe space including both two word clauses and words
  - Importance relates clause vecs and constituent words
- Sigmoidal Sentence Embeddings
  - Calculate document average word vector and covariance
  - For sentence, calculate each word's importance
  - Divide by double the sentence average
    - (Opt.) Ignore words in closest 20% of doc importances for sentence average
    - Corresponds closely to stop words
  - Weight by sigmoid of relative importances

Algorithms: Sentence Unembeddings

- Meaning Subtraction
  - $\text{vec(sentence)} = \Sigma\, w(\text{vec}_{\text{word\_n}}) * \text{vec}_{\text{word\_n}}$
  - Given a sentence vector and one subsentence vector, can calculate other subsentence vector
  - Assume $w(\text{vec}_{\text{other}})$ is the avg distance, solve for vec, repeat
  - Takes 3-5 iterations to converge to several decimal places
- Path-finding for Meaning Extraction
  - Calculate the remaining subsentence vector
  - If within m-cosine distance radius, return sentence
  - Find the new words closest to the subsentence vector
  - Enqueue the sentence with the closest words appended

Conclusion and Future Directions

- This technique should replace the tf-idf baseline
- Can global context help generate word vectors?
- Implications for how we process information
  - Appears to suggest we overvalue slightly more salient information when combining meanings
- Linguistic implications:
  - Where does syntax come into play?
  - Can a rule-based system restricting the subset of closest words that can be chosen as the next word generate grammatical sentences with the unembedding?
- Neurological implications: Can we measure the importance (salience) of words and sentences and relate them?