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 SoA 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 vcs: 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:
  - \[ \cos C = \frac{x^T y}{\sqrt{x^T x \cdot y^T y}} \]
- Wikipedia page for “green”
- Article about Stanford
- vec(“cardinal”) vs vec(“red”)
- \( d = 0.909 \)
- \( d = 0.943 \)

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
  - Corresponds closely to stop words
  - Weight by sigmoid of relative importances

Algorithms: Sentence Embeddings
- Meaning Subtraction
  - \( \text{vec(sentence)} = \Sigma w(\text{vec} \_\text{word}_n) \cdot \text{vec} \_\text{word}_n \)
  - Given a sentence vector and one subsentence vector, can calculate other subsentence vector
  - Assume \( w(\text{vec} \_\text{word}_n) \) 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 unembdding?
- Neurological implications: Can we measure the importance (salience) of words and sentences and relate them?