P&P 2

1. The distributional hypothesis states that the meaning of a word can be defined by its use and, therefore, it can be represented as a distribution of contexts in which the word occurs in a large text corpus. Name and describe four different types of contexts that can be used for this purpose.

- Unfiltered word windows: this type of context includes n words on either side of the lexical items as the context. For n=2 and the word "cat" in sentence "The large cat ate the fish", the context is [the 2, large 1, ate 1, fish 0].
- Filtered word windows: this type of context includes n words on either side of the lexical items as the context, but removes some words, such as function words. For n=2 and the word "cat" in sentence "The large cat ate the fish", the context is [large 1, ate 1, fish 0].
- Lexeme window (filtered or unfiltered): this type of context is the similar to the filtered word windows but uses stems of words. For n=2 and the word "cat" in sentence "The large cat ate the fish", the context is [large 1, eat 1, fish 0].
- Syntactic relations (dependencies): this type of context for a lexical item is the syntactic dependency structure it belongs to. For n=2 and the word "cat" in sentence "The large cat ate the fish", the context are: [large 1, eat 1], [large_mod 1, eat_subj 1], [large 1, eat+fish 1]

2. The contexts can be weighted using Pointwise Mutual Information (PMI). Explain how PMI is calculated and how individual probabilities are estimated from a text corpus, giving the formulae.

The point wise mutual information evaluate how much the actual co-occurrence of two events differs from their expected co-occurrence under the assumption of independence. Intuitively, PMI calculates the ratio between the frequency of context and word co-occurrence and the frequency of context and word co-occurrence if they are independent:

$$\text{PMI(w, c)} = \log \frac{P(w,c)}{P(w)P(c)} = \log \frac{P(c|w)P(w)}{P(w)P(c)} = \log \frac{P(c|w)}{P(c)}$$

This ratio tells us how much the actual co-occurrence deviates from the expectation under independence. This allows us to weight the context as PMI tells us how much the specific context c is informative about w compared to a baseline of independence. In practice PMI can be estimated with the following from a corpus,

$$\text{PMI}(w,c) = \log \frac{f(w,c)\sum_k f(c_k)}{f(w)f(c)}$$

Since $P(c) = \frac{f(c)}{\sum_k f(c_k)}$ and $P(c|w) = \frac{f(w,c)}{f(w)}$, and $f(w,c)$ is the frequency of word w in context c, $f(w)$ is the frequency of word w in all contexts, $f(c)$ is the frequency of context c. $\sum_k f(c_k)$ is a constant for the entire corpus, representing the total number of context instances. We calculate these frequencies from the text corpus.

3. Some words occur very rarely in the corpus. Show how this affects their PMI scores as contexts, and explain why this effect occurs.

For rare words their marginal probabilities $P(w)$ will be very low, leading to a very small denominator $P(w)P(C)$ that estimates the expected co-occurrence. As such even a small number of actual co-occurrence $P(w,c)$ would lead to an inflation of the PMI score due the very small denominator. This means that a high PMI score for a rare word-context pair might be a reflection of the rarity of the word, rather than actual semantic or syntactic relationship between the pair.

4. How are the clusters produced in the two experiments different with respect to the similarity they capture? What lexico-semantic relations do the clusters exhibit?

In experiment 1, the clusters captures taxonomic or categorical similarity. For example, "carriage" and "bike" are both hyponyms of transportation vehicles, "official" and "inspector" are both hyponyms of occupations, and "daughter" and "relative" are both hyponyms of relationships. Synonymy is also captured with words like "officer" and "policeman".

In experiment 2, the clusters captures thematic or contextual similarity. Lexico-semantic relations captured in these clusters include collocation of words in similar contexts (like "driver" and "highway" or "concert" and "singer"), thematic associations of words, and functional relations that describe objects and actions associated with activities (like "steering" for the first cluster and "research" in the third cluster).

5. The same clustering algorithm, K-means, was used in both experiments. What was different in the setup of the two experiments that resulted in the different kinds of similarity captured by the clusters? Give 2 such design choices, explain how each of these differed between the two experiments, and how this affects the resulting clusters in each experiment.

One design choice for the clustering algorithm is the feature representation used for representing each of the words. In experiment 1, feature representations derived based on contextual or semantic similarity is more likely used, as words that are hyponyms and synonyms are clustered together, suggesting that the features encode semantic similarity. In experiment 2, co-occurrence based feature representations are more likely used, as words are clustered based on thematic or contextual similarity. Words like "car" and "engine" are likely to co-occur in the same context, thus encoded similarly in their feature representations.

Another design choice may be based on the choice of the similarity metric. To capture semantic similarity the features representation needs to encode abstract relationships between words, and thus experiment 1 likely focused on higher dimensional representations of words where cosine similarity was used to calculate the feature similarities and capture deeper semantic relationships between the words. In experiment 2 for capturing contextual similarity, a simpler metric like Euclidean distance between the context word co-occurrence count can be used. The resulting clusters in this case, as mentioned in the previous question, captures the thematic similarity of the words.

6. Explain what is PP- attachment ambiguity using an example, and explain why it might be challenging for a syntactic parser. Outline how techniques from distributional semantics can be used in conjunction with a syntactic parser to help disambiguate prepositional phrase attachment ambiguities. Explain how such a system could be designed and how you would then use it to assign the correct parsing to a new sentence.

Prepositional Phrase (PP)-attachment ambiguity refers to the ambiguity that arise from a prepositional phrase in a sentence that could syntactically modify multiple part of the sentence. For example in the sentence, "She painted the vase with a brush", the PP could modify either "the vase" to suggest that the vase has a brush (which is syntactically valid), or modify "painted" to suggest that the act of painting was performed with the brush. This is a challenge to a syntactic parser as parsers typically rely on grammatical rules to resolve ambiguities, but in such ambiguities both interpretations can be syntactically valid and the true interpretation may depend on semantics or contexts. In the provided example, we can clearly deduce semantically that the PP attaches to "painted", but a syntactic parser that relies only on grammar rules would not be able to.

Distributional semantics relies on the distributional hypothesis of word meanings that says the context surrounding a given word provides information about its meaning, and thus distributional semantics construct representations of words in a high dimensional semantic space based on the usage of the word in a large corpora. These high dimensional representation of words, or embeddings, thus provide a method for evaluating the compatibility of the PP with the different attachments by measuring the similarity of the embeddings in the semantic space.

A system that could help disambiguate the PP attachment could:

1. Use a dependency parser like the Stanford parser to generate possible parse trees of the sentences.
2. Encode the sentence to embeddings using a distributional semantic model.
3. Compute similarity score between the PP and candidates. In our example this would be the similarity score between "the vase"-"with a brush" and "painted"-"with a brush".
4. The syntactic probabilities can then be combined with the semantic similarity scores to determine the more probable attachment. In our example the system should output the parse with "with a brush" attached to "painted".

7. The original skip-gram model learns dense word representations, i.e. *word embeddings*, by predicting a distribution of possible contexts for a given word. It thus treats the task as multiclass classification and uses the *softmax* function in its training objective. What problem arises with this model when it is being trained on a large text corpus and why?

By treating the task of learning the word embeddings as a multiclass classification task with the softmax function in its training objective, the objective of the skip-gram model is

$$\text{argmax} \prod_{(w_j,w_k)\in D} p(w_k|w_j) \Rightarrow \text{argmax} \prod_{(w_j,w_k)\in D} \frac{e^{c_k\cdot v_j}}{\sum_{i\in V} e^{c_i\cdot v_j}}$$

where for the word w_j indexed at j in the vocabulary, the model predicts word w_k indexed at k in the vocabulary. To compute the probability $p(w_k|w_j)$, the objective is then chosen as the softmax of the similarity between the target word embedding vector v_j and the context vector c_i. The problem with this objective is that the skip-gram model must have as many input and output dimensions as the vocabulary size. For a large text corpus, this vocabulary size can be very large and thus cause the computation of the softmax denominator that sums over the vocabularies to be very expansive.

8. How does skip-gram with negative sampling address the above problem? Briefly describe the intuition behind skip-gram with negative sampling, giving the formula for its training objective.

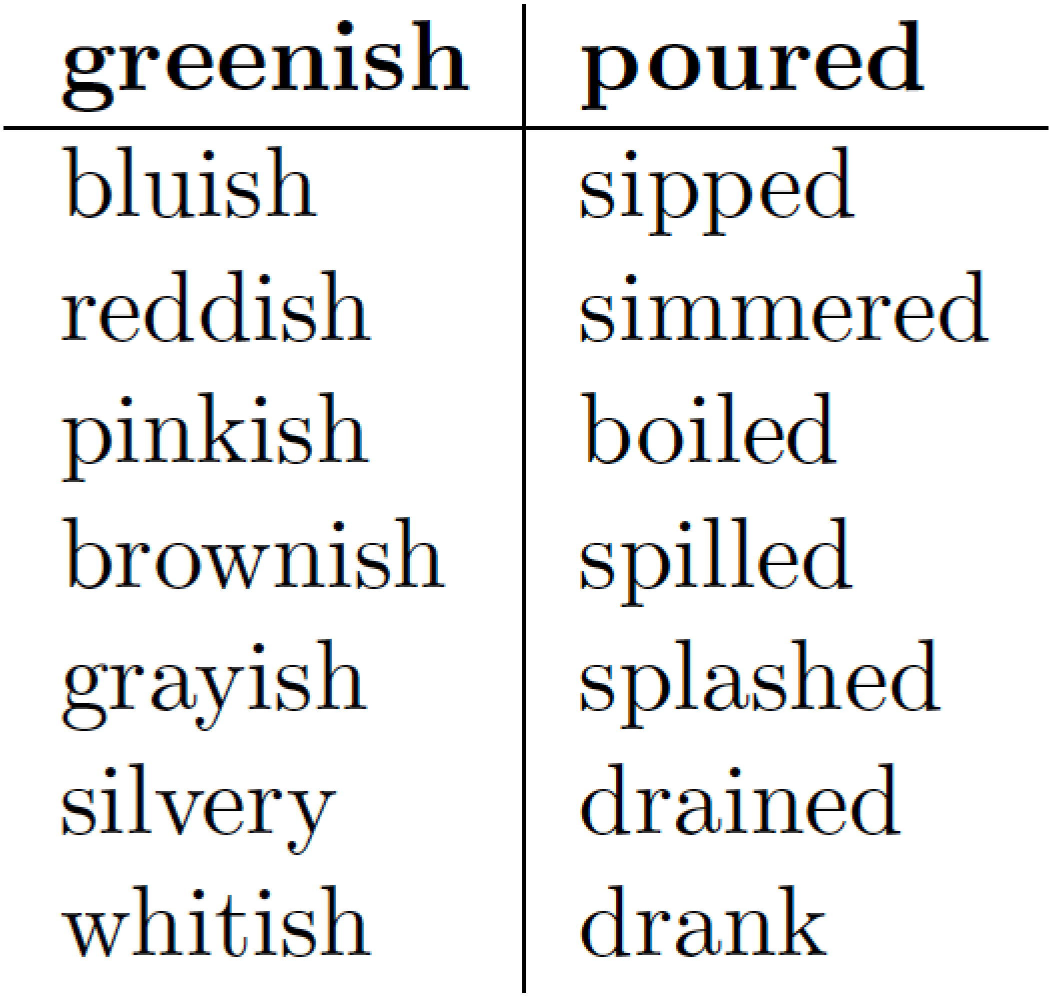
Skip-gram with negative sampling addresses the above problem by approximating the denominator of the softmax instead of explicitly calculating the summation over all vocabularies. For each target word and words in context seen as positive pair samples in the corpus during training of the skip-gram model, k samples are created from the target word and words from the vocabulary to represent negative samples. This allows for a reformulation of the problem from a multi-class problem that returns a probability distribution over the whole vocabulary into a binary classification problem that predicts if a given pair (w_j, w_k) is a pair from the context or a negative pair. With this in mind, the similarities modeled with dot products are converted into probabilities with the sigmoid function, and the training objective of the model becomes,

$$\begin{aligned} &\text{argmax} \prod_{(w_j,w_k)\in D_+} p(+|w_k, w_j) \prod_{(w_j,w_k)\in D_-} p(-|w_k, w_j) \\ \Rightarrow &\text{argmax} \sum_{(w_j,w_k)\in D_+} \log \frac{1}{1+e^{-c_k\cdot v_j}} + \sum_{(w_j,w_k)\in D_-} \log \frac{1}{1+e^{c_k\cdot v_j}} \end{aligned}$$

where D_+ and D_- now represents the positive and negative sample pairs. This simplification makes skip-gram with negative sampling computationally feasible for large corpora while retaining the ability to learn meaningful word embeddings.

Skip-gram word embeddings have two properties:

First property: They capture similarity in word meaning. The following examples show words most similar to *greenish* and *poured*, according to the skip-gram model.

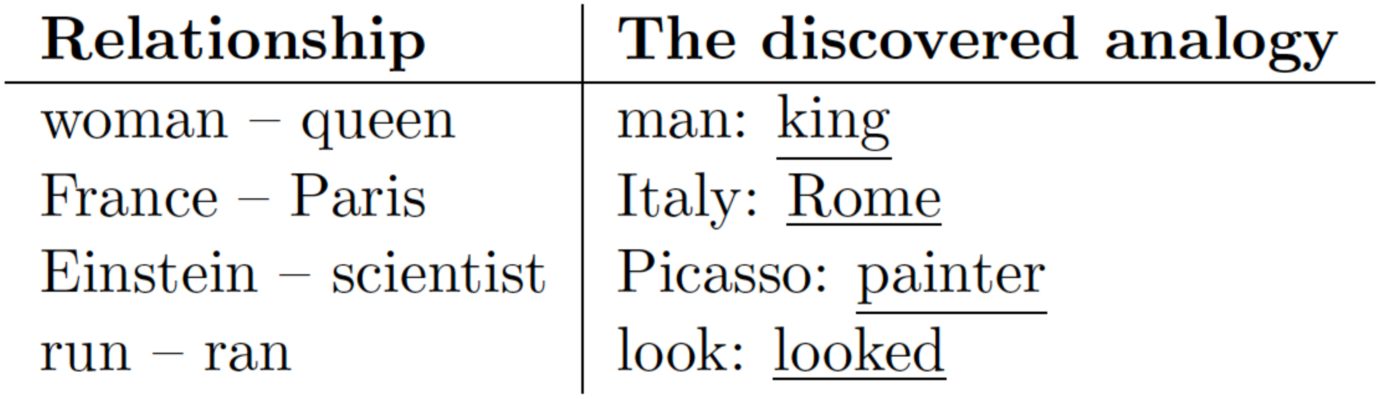


greenish	poured
bluish	sipped
reddish	simmered
pinkish	boiled
brownish	spilled
grayish	splashed
silvery	drained
whitish	drank

9. What aspects of word meaning do skip-gram word embeddings capture, as demonstrated by these examples? Describe two different aspects.

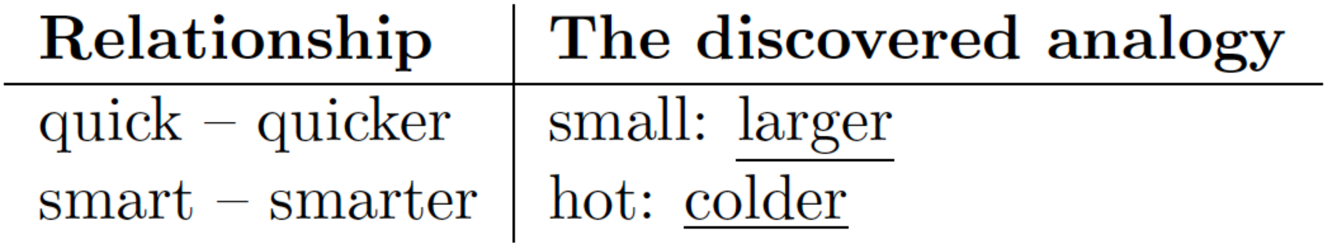
In the above examples, it is clear that the word embeddings capture semantic similarity and functional or contextual similarity. In the group "greenish", the word "silvery" with a different suffix other than "-ish" is also captured, indicating that semantic similarity is captured. Functional similarity can be demonstrated by the word group "pours", "spilled", "splashed", and "drained", all of which share close meanings related to actions of handling liquids. Word groups like "greenish", "bluish", and "reddish" are color related words that can be used in similar descriptive context or even replace one another, indicating the word embeddings' ability to capture contextual similarities.

Second property: Skip-gram word embeddings also capture analogy, as demonstrated below. The underlined words are automatically selected by the model; other words are provided as input.



Relationship	The discovered analogy
woman – queen	man: king
France – Paris	Italy: Rome
Einstein – scientist	Picasso: painter
run – ran	look: looked

The following examples show some of the system errors in the analogy task:



Relationship	The discovered analogy
quick – quicker	small: larger
smart – smarter	hot: colder

10. Explain briefly why these errors arise.

These errors in the analogy task occurs as a result of the limitations of skip-gram model's ability in capturing finer-grained semantic relationships in the word embeddings. As words and their context in corpuses are used for extracting word embeddings in the skip-gram model, words that often co-occur in same contexts (that may have more complicated relationships, such as opposites or comparatives) can be modeled closer in the embedding space. The above examples demonstrates exactly this phenomenon, where opposites like "small-larger" and "hot-colder" are closer in the embedding space due to shared context, despite their semantic relationships are different from that of "quick-quicker". This is demonstrates the limitations of these word embeddings in capturing non-linear relationships in the embedding space. Unlike examples like "$man - woman \approx king - queen$", nuanced relationships that may be non-linear such as comparatives or opposites are harder for the word embeddings to capture.