

*For each multiple-choice question below, mark the **single best** answer by completely filling in the circle.*

1) JSON is an example of

- ☐ A programming language
- ☒ A data format standard
- ☐ An unstructured data application
- ☐ None of the above

2) Which of the following transformations would **not** be made by a stemmer?

- ☐ going -> go
- ☐ goes -> go
- ☒ went -> go
- ☐ All of the above transformations could be made by a stemmer

3) Consider the following sentence:

A writer is a person who cares what words mean, what they say, how they say it.

Suppose we use a simple tokenizer that transforms to lowercase and removes punctuation. Which of the following is a sparse bag of words representation of the sentence?

- ☐ {a:1, writer:1, is:1, a:1, person:1, who:1, cares:1, what:1, words:1, mean:1, what:1, they:1, say:1, how:1, they:1, say:1, it:1}
- ☒ {a:2, writer:1, is:1, person:1, who:1, cares:1, what:2, words:1, mean:1, they:2, say:2, how:1, it:1}
- ☐ {writer:1, person:1, cares:1, words:1, mean:1, say:2}
- ☐ None of the above is a sparse bag-of-words representation of the sentence.

4) Which of the following are characteristics of applications built using the UIMA standard?

- ☐ Annotation-oriented processing of data streams
- ☐ Use XML for data communication
- ☐ Use a pipeline-like architecture where analyses engines may be chained together
- ☒ All of the above

- 5) Suppose you have the matrix  $V$  resulting from applying latent semantic analysis to a term-document matrix  $M$ . Consider a document  $d$  in the corpus that was used to create  $M$ .

Write a paragraph (about 4-6 sentences) explaining how you could use  $V$  to find the 5 documents in the corpus that are most similar to  $d$ . (Excluding  $d$  itself.) Make sure you clearly describe your notion of similarity.

Then describe in 3-5 sentences why this approach may work better for retrieving similar documents than using the term-document matrix  $M$  alone.

The matrix  $V$  obtained from LSA has  $m$  rows and  $p$  columns. Each column of the  $V$  transpose is a  $p$ -dimensional vector that represents a document from the corpus. We can start by identifying the column that corresponds to document  $d$ . We then use cosine similarity to determine how similar that column is to all of the other columns. The documents corresponding to the five most similar columns can be considered the most similar to  $d$ , with the exception of  $d$  itself (which will have a similarity of 1.0).

We could do the same by comparing columns of  $M$ , each of which represents a document. However, in this case, documents will only be considered similar if they contain identical words. By using the representation, documents are considered similar if they contain words from the same topic, even if those words differ from one document to the other. This may retrieve more relevant documents.