Leveraging Term Banks for Answering Complex Questions: A Case for Sparse Vectors

Peter D. Turney
Independent Researcher
This talk describes research conducted while I was employed at the
Allen Institute for Artificial Intelligence
2017

ALLEN INSTITUTE *for* ARTIFICIAL INTELLIGENCE

Outline

- Introduction
  - Answering multiple-choice science questions with unsupervised vector space models
- Related work
  - Past work with exam questions and past observations about sparsity and density
- Multivex
  - An algorithm for leveraging term banks with three types of vector spaces
- Experiments
  - Comparison with baselines and experiments with sparsity and density
- Trouble with embeddings
  - When sparsity is a good thing
- Future work and limitations
  - Next steps
- Conclusion
  - Advantages of term banks and sparse vectors

Introduction

Introduction

Motivation:

- Standard IR techniques cannot answer complex questions
- Standard KB techniques require expensive knowledge engineering
- Motivation is to cover the middle ground between IR and KB
- Intermediate level of question complexity
  - More complex than IR questions
  - Less complex than KB questions
- Intermediate level of resource requirements
  - More expensive resources than IR corpora
  - Less expensive resources than KB if-then rules and knowledge tables

Introduction

The middle ground:

- Use a *term bank* as an inexpensive resource for question answering
  - Assume questions are limited to a specific domain
  - Assume every specific domain has its own special vocabulary; its own *term bank*
  - Required resource is a *term bank* for the given specific domain
- Multivex uses three types of vector spaces constructed from a term bank
  - *Multivex* = multiple vector spaces
  - Given a term bank
  - Given a large corpus such that the terms in the term bank occur frequently
  - Build various vector spaces from the occurrences of the terms in the corpus

Introduction

- Restricted domain chosen in this case is science
  - Elementary (3rd to 5th) and middle (6th to 8th) grade levels
  - Inexpensive resource for domain is a term bank of 9,009 science terms
  - Questions are multiple-choice text-only (no diagrams) science questions from real exams

Which of the following statements best explains why earthquakes occur more frequently in California than in Massachusetts?
(A) The rock found in California is igneous, but the rock found in Massachusetts is sedimentary.
(B) California is located on the boundary of two crustal plates, but Massachusetts is not.
(C) The rock under California is soft, but the rock under Massachusetts is hard.
(D) California is located on a continental plate, but Massachusetts is not.

- Middle school (6th to 8th grade)
- Correct answer is (B)

Introduction

- Multivex: *multiple* unsupervised *vector* space models based on science terms
  - Intuition: for every question, there is a key science term linking the question to the best answer
  - Intuition is related to *lexical cohesion* in discourse semantics (Morris and Hirst 1991)
  - Look in term bank of 9,009 science terms for linking terms that provide lexical cohesion

Which of the following statements best explains why earthquakes occur more frequently in California than in Massachusetts?
(A) The rock found in California is igneous, but the rock found in Massachusetts is sedimentary.
(B) California is located on the boundary of two crustal plates, but Massachusetts is not.
(C) The rock under California is soft, but the rock under Massachusetts is hard.
(D) California is located on a continental plate, but Massachusetts is not.

- *Earthquake* is the key science term that links the question to the correct answer (B)
- Linking term need not appear in either question or solution

Introduction

Terminology space: *earthquake* has a high cohesion with question and (B)

Word space: the word *plates* often appears in the context *crustal* in sentences that contain *earthquake,* which supports answer (B)

Sentence space: answer (B) is similar to the kinds of sentences that occur in the sentence space for *earthquake*

Which of the following statements best explains why earthquakes occur more frequently in California than in Massachusetts?
(A) The rock found in California is igneous, but the rock found in Massachusetts is sedimentary.
(B) California is located on the boundary of two crustal plates, but Massachusetts is not.
(C) The rock under California is soft, but the rock under Massachusetts is hard.
(D) California is located on a continental plate, but Massachusetts is not.

- The three spaces all agree that the term *earthquake* provides a cohesive link between the question and answer (B)

Introduction

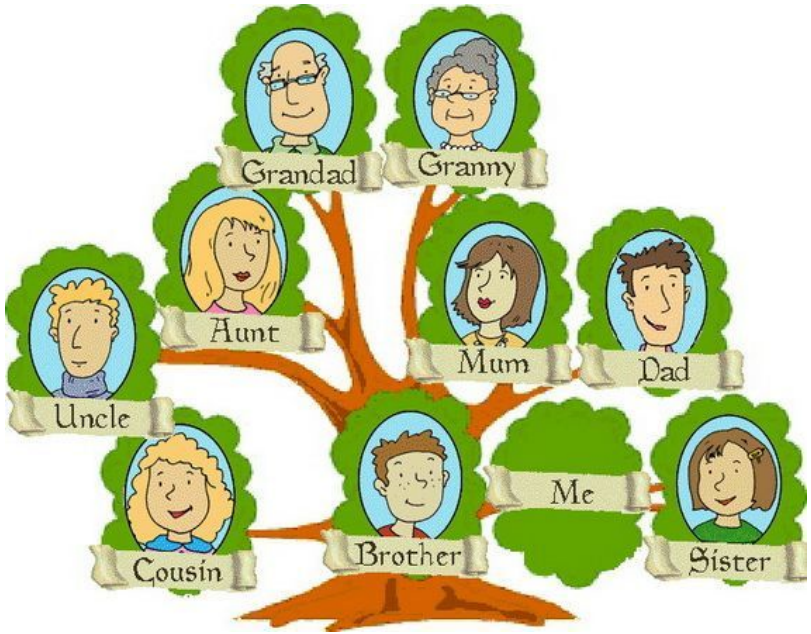
- Dense, low-dimensional embeddings versus sparse, high-dimensional vectors
  - Initial experiments with Multivex used dense, low-dimensional embeddings
  - Later experiments with Multivex used sparse, high-dimensional embeddings
  - Surprised to discover that sparse embeddings work best in Multivex
- Sparse vectors capture lexical cohesion better than dense vectors
  - Dense vectors are good for capturing the general sense of a word,
    but facts lie at the intersection of several word meanings
- Facts tend to be rare and specific
  - Which makes sparse vectors more appropriate when seeking facts
- Words are generalizations over many contexts
  - Which makes dense vectors more appropriate when modeling the meanings of words

Introduction

Two main results:

1. Leveraging term banks is an inexpensive way to answer complex questions in a restricted domain
2. Sparse vectors model facts better than dense vectors

Related Work



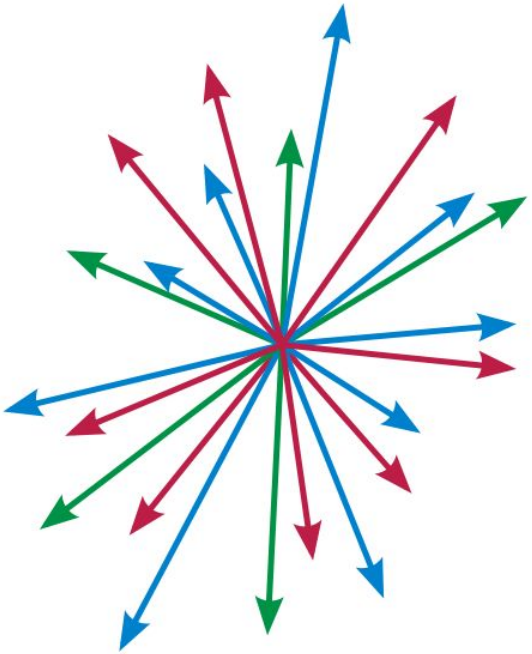
Related Work

- Past work with science exam questions
  - Khot et al. (2015) compared three different types of Markov Logic Networks (MLNs) for answering science exam questions; structured knowledge in the form of if-then rules
  - Clark et al. (2016) evaluated an ensemble of five solvers: three of the five were corpus-based, but the fourth used if-then rules and the fifth used tables; demonstrated that all five solvers made a significant contribution
  - Jauhar et al. (2016) represented science knowledge in a tabular form, where rows stated facts and columns imposed a parallel structure of types on the rows; best answer to a question was determined by the row and column that best supported one of the choices; trained a supervised log-linear model to score the choices
  - Khashabi et al. (2016) applied ILP to knowledge in a tabular form, using the same tables as Jauhar et al. (2016); ILP system performed multi-step inference by chaining together multiple rows from separate tables
- Common theme: expensive structured knowledge
  - If-then rules, knowledge tables

Related Work

- Dense, low-dimensional embeddings
  - Achieve good results on many tasks (Turney and Pantel, 2010)
  - Classical approach to embeddings is make word-context co-occurrence matrix and then apply dimensionality reduction (Landauer and Dumais, 1997)
  - More recent approach is to learn embeddings with a neural network (Mikolov et al. 2013a)
  - Baroni et al. (2014) describe classical approach as *context-counting* and neural approach as *context-predicting*, but Levy et al. (2014b) argue that both approaches learn same latent structure
- Sparse, high-dimensional vectors
  - Generally dense embeddings work better than sparse vectors on word similarity tasks (Landauer and Dumais, 1997; Turney and Pantel, 2010)
  - Levy and Goldberg (2014a) find sparse vectors superior in *"more semantic tasks"*
  - Toutanova et al. (2015) show sparse model is better than dense model in *knowledge bases for textual inference*

Multivex



Multivex

- **Input:** term bank, corpus, multiple-choice question
- **Output:** best choice for question, best term that links choice to question
- **Internal representation:** one terminology matrix, thousands of word matrices, thousands of sentence matrices

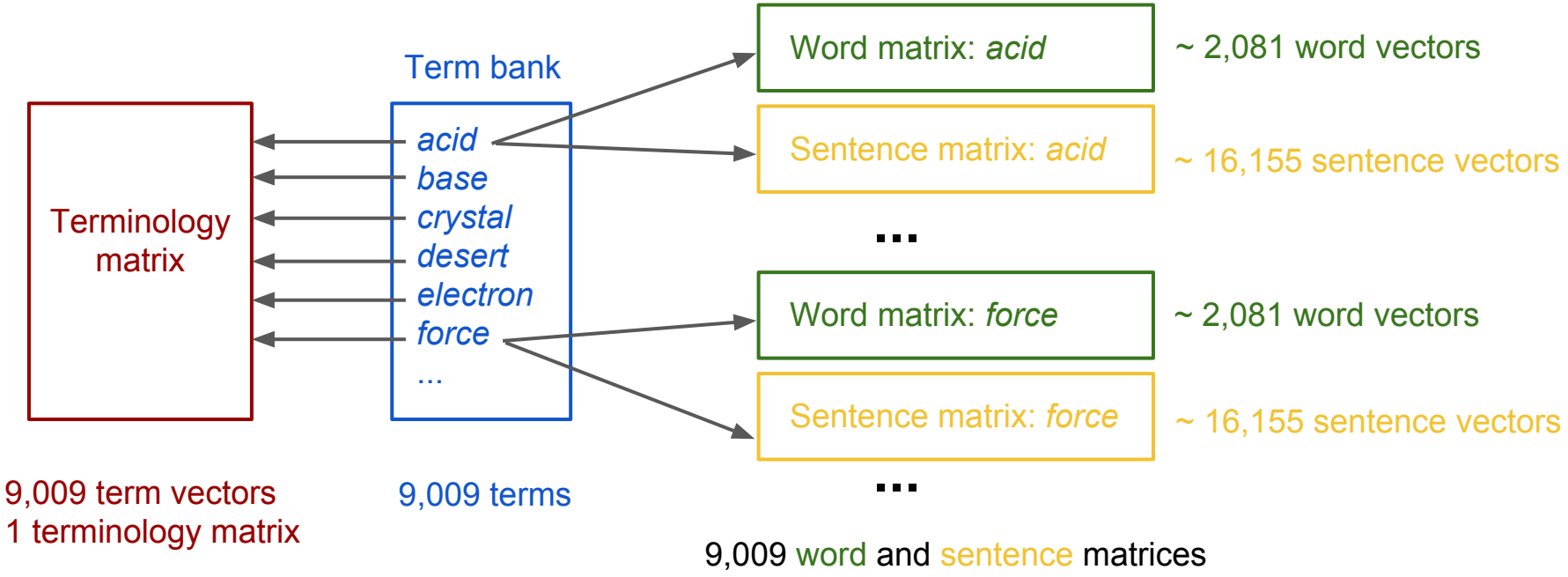
Matrices (Spaces)	Rows (Entities)	Columns (Features)
1 terminology matrix	9,009 science terms	22,767,476 unigrams and conjunctions
9,009 word matrices	2,081 words on average per matrix	millions of unigrams, bigrams, and trigrams
9,009 sentence matrices	16,155 sentences on average per matrix	millions of unigrams, bigrams, and trigrams

Multivex

Matrices (Spaces)	Rows (Entities)	Columns (Features)
1 terminology matrix	9,009 science terms	22,767,476 unigrams and conjunctions
9,009 word matrices	2,081 words on average per matrix	millions of unigrams, bigrams, and trigrams
9,009 sentence matrices	16,155 sentences on average per matrix	millions of unigrams, bigrams, and trigrams

- **Terminology matrix** is used to select candidate terms for given QA pair
- **Word matrix** and **sentence matrix** are selected based on given candidate term; word and sentence representations (meanings, senses) are conditional on chosen term
- The vector for a word in a QA pair (*plate, boundary, rock*) depends on the term (*earthquake*)
- A word (*plate*) can have up to 9,009 different vector representations (meanings, senses), one for each of the 9,009 word matrices
  - Related to Kilgarriff (1997), *I don't believe in word senses*
  - Word senses are *modulated* by choosing a science term as the *topic* of a QA pair

Multivex



Multivex

- Term bank
  - 9,356 terms from 52 K-12 science glossaries on web
  - 9,009 terms used in Multivex; terms with low frequency in corpus were dropped
  - Term bank is available from AI2 website
- Corpus
  - 280 GB of text, 50 billion tokens, collected by web crawler mostly from edu domain in 1990s
  - All markup removed and split into sentences with Stanford CoreNLP sentence segmenter
  - 1.75 billion English sentences
- Pseudo-documents
  - For each of the 9,009 terms, extract up to 50,000 sentences from the corpus containing term
  - Average of 16,155 sentences and 2,081 words per pseudo-document
  - Pseudo-document attempts to capture knowledge about each science term
  - The 9,009 pseudo-documents are available from AI2 website

Multivex

- Terminology Space
  - One matrix: 9,009 rows, one row for each science term
  - 22,767,476 columns, features derived from pseudo-documents for each science term
  - Features are unigrams and conjunctions of unigrams
  - Conjunctions occur together in a sentence in the given pseudo-document

Frequency	Feature
49,944	earthquake
4,149	flood
4,064	earthquake & flood
3,709	volcano
3,604	earthquake & volcano
3,254	earth
3,117	occur
3,062	earthquake & occur
2,969	natural
2,936	disaster

- Top ten most frequent unigrams and conjunctions of unigrams for the science term *earthquake*

Multivex

- Word space
  - 9,009 matrices, one for each science term
  - Each matrix is based on the pseudo-document for the given science term
  - Average 2,081 rows and millions of columns
  - Rows are word vectors, characterizing how a word behaves in the context of the science term
  - Each row corresponds to a word that appears in the pseudo-document for the science term
  - Columns are unigrams, bigrams, and trigrams near the given word in the pseudo-document
- Sentence space
  - 9,009 matrices, one for each science term
  - Each matrix is based on the pseudo-document for the given science term
  - Average 16,155 rows and millions of columns
  - Rows are sentence vectors, characterizing sentences that contain the given science term
  - Each row corresponds to a sentence that appears in the pseudo-document for the science term
  - Columns are unigrams, bigrams, and trigrams of sentences in the pseudo-document

Multivex

- **Terminology matrix:** given a term such as *earthquake*, what unigrams and conjunctions of unigrams should we expect to see in sentences (questions and answers) related to *earthquake*?
- **Word matrices:** given a term such as *earthquake*, what words tend to appear in sentences that contain earthquake and what kinds of contexts should we expect to see around these words?
- **Sentence matrices:** given a term such as earthquake, what sentences tend to contain earthquake and what n-grams should we expect to see in such sentences?

Multivex

Scoring QA pairs (q, a_i) with respect to candidate science terms t_j

- 4 steps, 2 scores per step, 8 scores total
- Only the best candidate science terms from one step pass on to the next step
  - First 4 subscores are based on terminology space
  - Next 2 subscores are based on word space
  - Last 2 subscores are based on sentence space
- Motivation for stepwise approach is speed
  - Calculations in terminology space are relatively fast; one single matrix is used for all subscores
  - Calculations in word space and sentence space require loading a new matrix for each new candidate science term

Multivex

Scoring QA pairs (q, a_i) with respect to candidate science terms t_j

- q = the question
- a_i = one of the four possible answers to q
- t_j = one of the terms in the term bank; a candidate linking term for q and a_i
- Each QA pair (q, a_i) is scored conditional on the candidate linking term t_j

Multivex

Step 1: terminology space with tf-idf weights

- Score 1.1: tf-idf weighted unigram match
  - weighted tf-idf match of unigrams in q and a_i with unigrams in t_j
- Score 1.2: tf-idf weighted conjunction match
  - weighted tf-idf match of conjunctions in q and a_i with conjunctions in t_j
- Only the top k_1 best t_j pass on to Step 2

Multivex

Step 2: terminology space with binary weights

- Score 2.1: binary unigram match
  - binary match of unigrams in q and a_i with unigrams in t_j
- Score 2.2: binary conjunction match
  - binary match of conjunctions in q and a_i with conjunctions in t_j
- Only the top k_2 best t_j pass on to Step 3

Multivex

Step 3: word space with tf-idf weights

- Score 3.1: word context match with same word
  - the average for each word *w* in *q* or a_i of the degree of match between the context around *w* in *q* or a_i and the context around *w* in t_j (zero if *w* is not in t_j)
- Score 3.2: word context match with different words
  - the average for each word *x* in *q* or a_i of the maximum degree of match between the context around *x* and the context around any word *y* in t_j
- Only the top k_3 best t_j pass on to Step 4

Multivex

Step 4: sentence space with binary weights

- Score 4.1: sentence whole match
  - treat the (q, a_i) pair as a sentence and find the best matching sentence in the sentence matrix for t_j
- Score 4.2: sentence subset match
  - find a large subset of the (q, a_i) pair that best matches a sentence in the sentence matrix for t_j
- Only the top k_4 best t_j pass on to the final result

Multivex

Summary of eight subscores

- Terminology space (1 matrix)
  - Score 1.1: tf-idf weighted unigram match
  - Score 1.2: tf-idf weighted conjunction match
- Terminology space (1 matrix)
  - Score 2.1: binary unigram match
  - Score 2.2: binary conjunction match
- Word space (9,009 matrices)
  - Score 3.1: word context match with same word
  - Score 3.2: word context match with different words
- Sentence space (9,009 matrices)
  - Score 4.1: sentence whole match
  - Score 4.2: sentence subset match

Multivex

- Final score for the QA pair (q, a_i) is the average of the eight subscores, given the top science term t_j selected by the four steps
  - The four terminology matrix scores ensure that the text in q and a_i is similar to the text in the pseudo-document for t_j
  - The two word matrix scores ensure that the words in q and a_i have contexts around them that are similar to the contexts around the words in the pseudo-document for t_j
  - The two sentence matrix scores ensure that the (q, a_i) pair, treated as a sentence, is similar to some of the sentences in the pseudo-document for t_j

Multivex

- The science term t_j is intended to capture the topic of the QA pair, to provide lexical cohesion between q and a_i
  - If a_i is the correct answer, there should be a science term t_j that can link a_i to q
- However, if there is a science term t_j that can link a_i to q, that does not necessarily mean that a_i is the right answer
  - Lexical cohesion is necessary for a good answer but not sufficient

Experiments

Experiments

- Summary of science exam question sets
  - We used train and development subsets while developing Multivex
  - We used test subsets for following experiments
  - Public questions are available for download from AI2 website

Questions	Train	Dev	Test	Total
Public Elementary	432	84	339	855
Public Middle	293	65	282	640
Licensed Elementary	574	143	717	1434
Licensed Middle	1581	482	1631	3694
All Questions	2880	774	2969	6623

Experiments

- Comparison of Multivex with Lucene, SVD, and Word2vec
  - All deltas from Multivex are statistically significant, Fisher's Exact Test, 95% confidence
  - Lucene: IR baseline using same corpus as Multivex
  - SVD 1, SVD 2: two different embeddings using Singular Value Decomposition
  - Word2vec 1, Word2vec 2: two different uses of Word2vec vectors trained with Google News

Algorithm	Public Elementary	Public Middle	Licensed Elementary	Licensed Middle	All Test Questions	Vector Type	Delta
Multivex	59.7	60.6	51.1	49.0	51.8	sparse	0.0
Lucene	55.8	52.5	48.7	47.3	49.1	sparse	−2.7
SVD 1	55.5	51.5	46.8	45.6	47.6	dense	−4.3
SVD 2	56.2	51.8	48.3	46.9	48.8	dense	−3.1
Word2vec 1	49.9	49.7	41.7	42.2	43.7	dense	−8.2
Word2vec 2	51.9	52.6	45.3	44.8	46.5	dense	−5.4

Experiments

- Comparison of Multivex with Lucene
  - Multivex is significantly better than Lucene (IR baseline)
  - Lucene is a tough baseline
  - None of the individual systems in Clark et al. (2016) surpassed Lucene; only an ensemble of five systems was better

Algorithm	Public Elementary	Public Middle	Licensed Elementary	Licensed Middle	All Test Questions	Vector Type	Delta
Multivex	59.7	60.6	51.1	49.0	51.8	sparse	0.0
Lucene	55.8	52.5	48.7	47.3	49.1	sparse	−2.7
SVD 1	55.5	51.5	46.8	45.6	47.6	dense	−4.3
SVD 2	56.2	51.8	48.3	46.9	48.8	dense	−3.1
Word2vec 1	49.9	49.7	41.7	42.2	43.7	dense	−8.2
Word2vec 2	51.9	52.6	45.3	44.8	46.5	dense	−5.4

Experiments

- Comparison of Multivex with SVD and Word2vec
  - Sparse, high-dimensional vectors are better than dense, low-dimensional vectors
  - Only the terminology space was made dense, to simplify interpretation of results
  - Four of eight subscores are based on terminology space
  - Terminology space narrows choice of science term down to four terms out of 9,009

Algorithm	Public Elementary	Public Middle	Licensed Elementary	Licensed Middle	All Test Questions	Vector Type	Delta
Multivex	59.7	60.6	51.1	49.0	51.8	sparse	0.0
Lucene	55.8	52.5	48.7	47.3	49.1	sparse	−2.7
SVD 1	55.5	51.5	46.8	45.6	47.6	dense	−4.3
SVD 2	56.2	51.8	48.3	46.9	48.8	dense	−3.1
Word2vec 1	49.9	49.7	41.7	42.2	43.7	dense	−8.2
Word2vec 2	51.9	52.6	45.3	44.8	46.5	dense	−5.4

Experiments

- Ablating subscores from Multivex
  - Delta = drop in accuracy when given subscore is removed

Step	Ablated Subscore	Delta
1.1	tf-idf weighted unigram match	+0.2
1.2	tf-idf weighted conjunction match	−2.1
2.1	binary unigram match	+0.2
2.2	binary conjunction match	−1.9
3.1	context match with same word	−0.6
3.2	context match with different words	−0.8
4.1	sentence whole match	−1.3
4.2	sentence subset match	−0.5

Experiments

- Ablating subscores from Multivex
  - Impact of the two unigram match subscores is negligible
  - Irrelevant given conjunction match subscores

Step	Ablated Subscore	Delta
1.1	tf-idf weighted unigram match	+0.2
1.2	tf-idf weighted conjunction match	−2.1
2.1	binary unigram match	+0.2
2.2	binary conjunction match	−1.9
3.1	context match with same word	−0.6
3.2	context match with different words	−0.8
4.1	sentence whole match	−1.3
4.2	sentence subset match	−0.5

Experiments

- Ablating subscores from Multivex
  - Most helpful subscores are conjunction matches
  - Conjunction match: two words that appear together in a sentence in a term's pseudo-document
  - Pseudo-document for *earthquake* → *crustal & plate*

Step	Ablated Subscore	Delta
1.1	tf-idf weighted unigram match	+0.2
1.2	tf-idf weighted conjunction match	−2.1
2.1	binary unigram match	+0.2
2.2	binary conjunction match	−1.9
3.1	context match with same word	−0.6
3.2	context match with different words	−0.8
4.1	sentence whole match	−1.3
4.2	sentence subset match	−0.5

Experiments

- Ablating subscores from Multivex
  - Next most helpful subscore is sentence whole match
  - Sentence whole match: match of unigrams, bigrams, and trigrams in a sentence in the science term's pseudo-document with unigrams, bigrams, and trigrams in the given QA pair

Step	Ablated Subscore	Delta
1.1	tf-idf weighted unigram match	+0.2
1.2	tf-idf weighted conjunction match	−2.1
2.1	binary unigram match	+0.2
2.2	binary conjunction match	−1.9
3.1	context match with same word	−0.6
3.2	context match with different words	−0.8
4.1	sentence whole match	−1.3
4.2	sentence subset match	−0.5

Experiments

- Varying parameters
  - Varying k_1, k_2, k_3, k_4 in Multivex
  - k_i is the number of the top terms that are passed on from Step *i* to Step *i*+1

Number of Top Terms				All Test Qs	
Step 1	Step 2	Step 3	Step 4	Score	Time
5	2	1	1	51.1	2.8
10	4	1	1	51.8	5.0
20	8	2	1	51.9	10.4
40	16	4	1	51.9	20.9

Experiments

- Varying parameters
  - Varying k_1, k_2, k_3, k_4 in Multivex
  - Default settings (used in preceding experiments) are $k_1 = 10$ $k_2 = 4$, $k_3 = 1$, $k_4 = 1$
  - $k_2 = 4 \rightarrow$ four terms are passed on from terminology space to word space
  - $k_3 = 1 \rightarrow$ one term is passed on from word space to sentence space

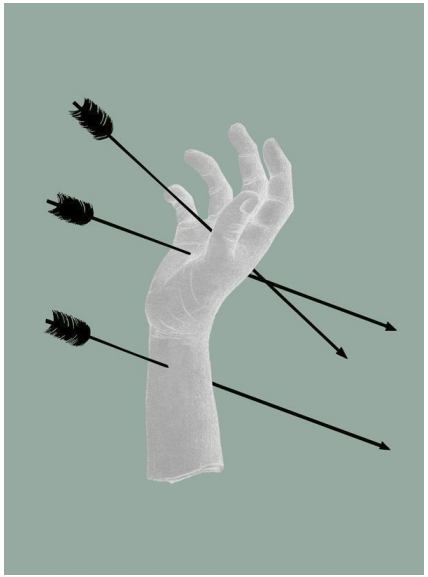
Number of Top Terms				All Test Qs	
Step 1	Step 2	Step 3	Step 4	Score	Time
5	2	1	1	51.1	2.8
10	4	1	1	51.8	5.0
20	8	2	1	51.9	10.4
40	16	4	1	51.9	20.9

Experiments

- Varying parameters
  - Time in seconds to answer one question (score four choices)
  - Default setting gives reasonable speed with negligible loss in score

Number of Top Terms				All Test Qs	
Step 1	Step 2	Step 3	Step 4	Score	Time
5	2	1	1	51.1	2.8
10	4	1	1	51.8	5.0
20	8	2	1	51.9	10.4
40	16	4	1	51.9	20.9

Trouble with Embeddings



Trouble with Embeddings

- Multivex works better with sparse, high-dimensional vectors
  - Performance drop when using SVD embeddings or Word2vec embeddings
- Preceding table shows tf-idf weighted conjunction match is the most helpful of the eight scores
  - tf-idf weighted conjunction match uses terminology matrix
  - 22,767,476 columns in the terminology matrix
  - 22,505,565 are conjunctions — 98.8% (remainder are unigrams)
  - Pseudo-document frequency of conjunction features ranges from 1 to 4,292, with median of 1
  - Conjunction features have a very long tail of rare events
  - Low-dimensional embeddings smooth away these rare events
- Facts are sparse, rare events: a randomly generated assertion is most likely to be false

Future Work and Limitations

Future Work and Limitations

- Focus of this research has been multiple-choice questions, but Multivex could be extended to direct-answer questions
  - The sentence matrices could be used to generate as set of candidate direct answers
- Multivex is unsupervised, but supervision may improve test scores
  - Could use machine learning to generate scores from vectors
  - Supervised deep learning with attention model might be able to focus on rare events
- Multivex uses unigrams, bigrams, trigrams, and conjunctions of unigrams
  - Could benefit from more complex features, such as part-of-speech tags and semantic relations
- Beyond term banks, other inexpensive resources could be used
  - Most of the science glossaries we used included definitions of the terms, but we did not use the definitions in Multivex
  - Definitions for science terms might yield improved vectors

Conclusion

Conclusion

- A term bank is an inexpensive resource for QA with complex questions
  - A domain-specific term bank is a relatively light-weight requirement, compared to if-then rules or knowledge tables
  - A term bank provides a way to measure lexical cohesion
  - The output term provides additional information, beyond simply choosing the correct answer
  - The term might be used to help a student search for more information about the question
  - Being able to identify the topic of a question is the first step towards deeper understanding
- Sparse, high-dimensional vectors are well-suited to QA with complex questions
  - Word meanings are distributional and general but facts are intersections of word meanings
  - Facts tend to be rare and specific
- As QA systems mature, research will shift from word meanings to sentence meanings
  - This will require a shift from dense embeddings to sparse vectors
  - Words are repeated but most sentences are unique