Did well:

it is a figure of that	-one valid word w/ stopwords
visualization	
computing architecture	
computer technology	
professor informatics engineering	
user interface software develop	
it is what it is	-only stopwords
each word occurs at the same time	-valid words w/ stopwords
!?!?!?!?!?!??!?!	-invalid

Did poorly:

uci undergraduate artificial intelligence research
computer science software engineering

- These words frequently appeared in many documents, so computing tf-idf at times for these took quite some time. Pre indexing tf-idf significantly reduced time.

research natural language processing machine learning
uci professor artificial intelligence advanced class

- There was no document which contains these all words, so I got no result from it. Implemented a function when the original input query can not be AND searched, the program will pop the last word from the query and continue until it gets a result.

internet tech internet

- the same words appeared in a query caused doubled calculation on the term weight. By using a tokenizer for an input query, the program computes only unique words in a query.

Cristina lopes

- In M2, results are mainly news, but by using tf-idf, the program returned profiles as well thus more effective.

master of software engineering

- In M2, results are not effective, since the second and the third ranks didn't contain master of software engineering. By using a kind of normalization, one of the relevant results is ranked up.

software development

- previous results contained publications with unbalanced term frequencies. By using a kind of normalization, the relevant results are ranked up.

informatics and engineering are the process of making something useful when it comes to computer technology

- This input was too long to AND search but improved by popping the last word if necessary.

Please note that some of the very old tests below reflect different textbooks that may define some things differently than does your current textbook

- Takes time each query takes 30-50 ms, so this still cannot be handled within 300 ms limit. It's very possible if I make a complete weight index.