Question analysis in IBM Watson

Paper Presentation

Antoni Maciąg, May 2022

The task

- Not strictly defined - given a question, identify key elements that will be helpful when looking for the answer.
- Here the approach consists in classifying the question and looking for 3 entities: focus, LAT and Question Sections.
- These problems are somewhat related to each other.
- We will look at what they are, how to discern them, what problems they present, and what they are useful for, as well as evaluate the solutions applied in Watson.

Focus

- The reference to the answer, contained in the question.
- ‘John the Savage falls in love with Lenina Crowne in this Aldous Huxley ***classic’***.
- The part most specifically referring to the answer, and hinting at its type: ***headword***.
- Basic detection looks for phrases of these types: noun phrase with ‘this’, ‘this’ as a pronoun, whole question being a noun phrase, pronouns, possessive pronouns, ‘one’.
- Sometimes none of them applies.

Problems with focus

- Headword detection (e.g. may not directly follow ‘this’).
- Reliance on correct parsing.
- There may be many pronouns (selecting unbound pronouns without co-references):

  ‘An April 1997 auction of Clyde Barrow’s belongings raised money to fund moving his grave next to hers.’

- More on that in the next part.

LAT

- Lexical Answer Type.
- Term in the question that indicates the type of the answer: ‘writer’, ‘they’, ‘region’.
- Baseline approach: headword is LAT.

Improvements in LAT detection

- Conjunction
- LAT=X if focus is '[one|name|type] of X'
- LAT=X if focus is '[name|word|term] for X'
- If no focus detected but category is a noun phrase, take its headword as LAT.
- Detecting subclass relations in focus ('Fars is famous for this **variety** of wine').
- Detect set membership relations ('Out of the **states in Orwell's <<1984>>**, this is the one the protagonist lives in').
- Determine the roles played by the focus ('Before becoming a **writer**, he earned his living as a truck driver in Silesia').

Improvements in LAT detection

- Learn from previous (question, answer) paris in the same category (e.g. “COUNTRIES’ FOODS: Bubble tea, stinky tofu” vs “COUNTRIES’ FOODS: Taiwan”).
- Use multiple LAT detectors with confidence estimates (very Watson thing to do).

Problems with LAT

- Single-word vs multi-word (‘this French president’, ‘this vice-president’). Solution: subclass detection, more subtly: infrequent sense detection (‘Greek fire’, ‘prime minister’, ‘first secretary’).
- LAT detection in category. Conditions:
- If there is a LAT in the question, the category should be consistent with it (EMPRESSES: *‘She* introduced new characters specifically for spelling her name‘, vs EMPRESSES: *‘This* was the profession of Empress Theodora before ascending the throne’).
- It should not appear in the question (take ‘LITERARY DEBUTS: Joseph Heller’ vs ‘LITERARY DEBUTS: Catch-22’).

LAT and focus evaluation

- Focus not evaluated independently.
- Standard F1 used, score: 0.796. Precision: 0.829, recall: 0.766.
- Detected LATs are fairly reliable, but many go undetected.

Note on relation detection

- Relation extractors exist.
- They might not include or be sensitive to relations most common in Jeopardy!, like relations in time and space, alternate names.
- The authors improved this, using (among others) the previously mentioned semantic frames for relation detection.

Question Classification

- Questions in Jeopardy! Have certain frequent types - it's good to identify them and apply specialized ways of answering.
- The task consists in determining the type correctly.
- Types (QClasses): definition, category relation ('former PMs: Thatcher, Blair'), FITB, abbreviation, puzzle ('Before & after: venetian traveler who's a short-sleeve top with a collar'), etymology, verb, translation, number, bond ('strange, charm, bottom'), multiple choice, date.

QClass detection

- Puzzle, bond, FITB, multiple choice - by regular expressions.
- Number, date - when the LAT suggests them.
- Definition - by key phrases in questions and categories. This does not work very well but turns out not to be a problem because standard methods work well for definition questions.
- Category-relation - if there is no focus.
- Etymology, translation: pattern matching of syntax, e.g. for translation: “From <language> for <phrase>, <focus> … ”
- Abbreviation: rules for syntax, frames and some other minor improvements.
- Verb: definition with infinitive or focus is the object of “do”.
- A question can have multiple QClasses. However, some are mutually incompatible.

QClass detection evaluation

- Percentage of correct answers improved by 2.9%. Detection scores:

QClass	*Precision*	*Recall*	F_1
DEFINITION	0.508	0.487	0.497
CATEGORY-RELATION	0.644	0.806	0.716
FITB	0.676	0.711	0.693
ABBREVIATION	0.728	0.806	0.765
PUZZLE	0.977	0.525	0.683
ETYMOLOGY	0.886	0.600	0.716
VERB	0.796	0.811	0.804
TRANSLATION	0.885	0.590	0.708
NUMBER	0.842	0.471	0.604
BOND	1.000	0.652	0.789
MULTIPLE-CHOICE	0.650	0.684	0.667
DATE	0.692	0.818	0.750
All	0.646	0.629	0.637

Question Sections

- Annotations over parts of questions and categories that indicate what role the part plays in the interpretation of the question.
- Examples: lexical constraint, abbreviation, subquestion, McAnswer, text to complete in FITB.
- Task: recognize them.
- Solution: checking the frame and text against handwritten rules or regular expressions.

Applications

- Focuses: finding supporting passages for the answer. The answer suggested by the passage should be consistent with the focus.
- LATs: validating candidate answers by type. More on this later in Przemysław's and Krzysztof's presentations.
- QClasses: applying different answering techniques.
- QSections: determining QClasses, imposing lexical constraints, decomposing into subquestions.

Note on implementation

- Prolog used to write down the detection rules.
- Authors stress that this was very convenient and also resulted in efficiency gains over the previous approach, which was “custom pattern-matching frameworks” (but no further details, like e.g. the programming language, were provided).
- 6000 Prolog clauses used in total for tasks mentioned in this presentation.

Thank you