AnswerFinder at TREC 2005
DIEGO MOLLÁ AND MENNO VAN ZAANEN
Centre for Language Technology, Macquarie University, Sydney, Australia
{diego, menno}@ics.mq.edu.au

ABSTRACT
AnswerFinder has been completely redesigned for TREC 2005. The new architecture allows for the fast development of question-answering systems for their deployment in the TREC tasks and applications. The modules use XML to express the services they provide, and they can be queried with XML for their services. In terms of the QA method, a major difference with respect to previous years is the use of graph-based methods to compute the answerhood of a sentence and pin-point the answer. The system uses a set of graph-based patterns that are learned automatically.

1 Introduction
AnswerFinder is a research-oriented question answering system. It has been completely redesigned and reimplemented since last year:
- Incorporates symbolic information.
- Uses automatically induced structural semantic information.
- Incorporates flat logical forms and logical graphs (LG).

2 System overview

3 Logical Graphs (LG)
- AnswerFinder uses a shallow representation of the semantics of the question and the sentences to find relevant sentences.
- This year, LGs were tested. They are not only used to score sentences, but also to find actual answers.
- LGs are similar to Sowa’s (1979) conceptual graphs, but LGs do not attempt to encode the full semantics.
- LGs are computed from the logical forms, which were used in last years competition (Molla and Gardiner, 2004).
- Logical forms avoid representation of well-known problematic concepts such as quantification, plurality, tense, and aspect.
- LGs are directed, bipartite graphs with two types of vertices: concepts and relations.
- Concepts: For example, objects, dog, table, events and states can, love, and properties red, quick.
- Relations: Relations act as links between concepts. Examples of relations would be grammatical roles and prepositions. We use relation labels that are relatively close to the syntactic level, such as subject, object, etc. In fact, we use numbers that represent verb arguments. 1 indicates the first argument of a verb (that is, what is usually a subject). The relation 2 indicates the second argument of a verb (direct object), and so forth.

3.1 Problem formulation
The problem is to find the complete answer. Answer expansion is then performed by adding all concepts that are acceptable from the answer.

4 Logical Graph Rules (LGR)
- LGs can be used to extract precise answers from sentences by applying the question LG to the sentence LG.
- Computing graph overlap and paths between subgraphs indicates relevance and finds possible answers. (Molla and van Zaanen, 2005)
- Each rule ⊃ contains three components:
  - , overlap between a question and its answer sentence;
  - , path between the overlap and the answer in the sentence;
  - , graph representing the exact answer.
- LGs are learned from questions and sentences that are annotated with the answer.

5 Graph-based Question Answering
- All LGRs are applied to the sentences. A LGR triggers if , in a subgraph of the question. The LG of the question is then extended with .
- The expanded LG is used to compute overlap with the sentences.
- The new LG has a higher overlap with LGs of relevant sentences and indicates possible answers.
- Possible answers are ranked using , where and . The size of the overlap between the expanded LG of the question and the sentence:
- The size of a graph overlap is computed as the weighted sum of all concepts and relations in the overlap. The weight of a concept or relation in the overlap is determined using a variant of the inverse Document Frequency (IDF) measure used in document retrieval. 

6 Parameters

7 Results

8 Conclusions
AnswerFinder is completely redesigned and reimplemented. The system is highly modular and therefore easy to extend.
LGs and LGRs seem to work well in theory, but the implementation needs to be improved.
Additional implementation problems need to be fixed.

Acknowledgements
This research is funded by the Australian Research Council, ARC Discovery Grant no. DP1403078.

DIEGO MOLLÁ AND MENNO VAN ZAANEN
Centre for Language Technology, Macquarie University, Sydney, Australia
{diego, menno}@ics.mq.edu.au

ABSTRACT
AnswerFinder has been completely redesigned for TREC 2005. The new architecture allows for the fast development of question-answering systems for their deployment in the TREC tasks and applications. The modules use XML to express the services they provide, and they can be queried with XML for their services. In terms of the QA method, a major difference with respect to previous years is the use of graph-based methods to compute the answerhood of a sentence and pin-point the answer. The system uses a set of graph-based patterns that are learned automatically.

1 Introduction
AnswerFinder is a research-oriented question answering system.
- It has been completely redesigned and reimplemented since last year.
- Incorporates symbolic information.
- Uses automatically induced structural semantic information.
- Incorporates flat logical forms and logical graphs (LG).

2 System overview

3 Logical Graphs (LG)
- AnswerFinder uses a shallow representation of the semantics of the question and the sentences to find relevant sentences.
- This year, LGs were tested. They are not only used to score sentences, but also to find actual answers.
- LGs are similar to Sowa’s (1979) conceptual graphs, but LGs do not attempt to encode the full semantics.
- LGs are computed from the logical forms, which were used in last years competition (Molla and Gardiner, 2004).
- Logical forms avoid representation of well-known problematic concepts such as quantification, plurality, tense, and aspect.
- LGs are directed, bipartite graphs with two types of vertices: concepts and relations.
- Concepts: For example, objects, dog, table, events and states can, love, and properties red, quick.
- Relations: Relations act as links between concepts. Examples of relations would be grammatical roles and prepositions. We use relation labels that are relatively close to the syntactic level, such as subject, object, etc. In fact, we use numbers that represent verb arguments. 1 indicates the first argument of a verb (that is, what is usually a subject). The relation 2 indicates the second argument of a verb (direct object), and so forth.

3.1 Problem formulation
The problem is to find the complete answer. Answer expansion is then performed by adding all concepts that are acceptable from the answer.

4 Logical Graph Rules (LGR)
- LGs can be used to extract precise answers from sentences by applying the question LG to the sentence LG.
- Computing graph overlap and paths between subgraphs indicates relevance and finds possible answers. (Molla and van Zaanen, 2005)
- Each rule ⊃ contains three components:
  - , overlap between a question and its answer sentence;
  - , path between the overlap and the answer in the sentence;
  - , graph representing the exact answer.
- LGs are learned from questions and sentences that are annotated with the answer.

5 Graph-based Question Answering
- All LGRs are applied to the sentences. A LGR triggers if , in a subgraph of the question. The LG of the question is then extended with .
- The expanded LG is used to compute overlap with the sentences.
- The new LG has a higher overlap with LGs of relevant sentences and indicates possible answers.
- Possible answers are ranked using , where and . The size of the overlap between the expanded LG of the question and the sentence:
- The size of a graph overlap is computed as the weighted sum of all concepts and relations in the overlap. The weight of a concept or relation in the overlap is determined using a variant of the inverse Document Frequency (IDF) measure used in document retrieval. 

6 Parameters

7 Results

8 Conclusions
AnswerFinder is completely redesigned and reimplemented. The system is highly modular and therefore easy to extend.
LGs and LGRs seem to work well in theory, but the implementation needs to be improved.
Additional implementation problems need to be fixed.

Acknowledgements
This research is funded by the Australian Research Council, ARC Discovery Grant no. DP1403078.