ABSTRACT
Question answering (QA) over text passages is a problem of long-standing interest in information retrieval. Recently, the conversational setting has attracted attention, where a user asks a sequence of questions to satisfy her information needs around a topic. While this setup is a natural one and similar to humans conversing with each other, it introduces two key research challenges: understanding the context left implicit by the user in follow-up questions, and dealing with ad hoc question formulations. In this work, we demonstrate CROWN (Conversational passage ranking by Reasoning Over Word Networks): an unsupervised yet effective system for conversational QA with passage responses, that supports several modes of context propagation over multiple turns. To this end, CROWN first builds a word proximity network (WPN) from large corpora to store statistically significant term co-occurrences. At answering time, passages are ranked by a combination of their similarity to the question, and coherence of query terms within: these factors are measured by reading off node and edge weights from the WPN. CROWN provides an interface that is both intuitive for end-users, and insightful for experts for reconfiguration to individual setups. CROWN was evaluated on TREC CaST data, where it achieved above-median performance in a pool of neural methods.

CCS CONCEPTS
• Information systems → Question answering.

KEYWORDS
Conversational Search, Conversational Question Answering, Passage Ranking, Word Networks

1 INTRODUCTION

Motivation. Question answering (QA) systems [13] return direct answers to natural language queries, in contrast to the standard practice of document responses. These crisp answers are aimed at reducing users’ effort in searching for relevant information, and may be in the form of short text passages [5], sentences [19], phrases [3], or entities from a knowledge graph [10]. In this work, we deal with passages: such passage retrieval [17] has long been an area of focus for research in information retrieval (IR), and is tightly coupled with traditional text-based QA [18]. Passages are one of the most flexible answering modes, being able to satisfy both objective (factoid) and subjective (non-factoid) information needs succinctly.

Of late, the rise of voice-based personal assistants [6] like Siri, Cortana, Alexa, or the Google Assistant has drawn attention to the scenario of conversational question answering (ConvQA) [4, 15]. Here, the user, instead of a one-off query, fires a series of questions to the system on a topic of interest. Effective passage retrieval often holds the key to satisfying such responses, as short passages or paragraphs are often the most that can be spoken out loud, or displayed in limited screen real estate, without sacrificing coherence. The main research challenge brought about by this shift to the conversational paradigm is to resolve the unspoken context of the follow-up questions. Consider our running example conversation below, that is a mix of factoid (turns 1, 2 and 3), and non-factoid questions (turns 4 and 5). Answers shown are excerpts from top paragraphs retrieved by an ideal passage-based ConvQA system.

Question (Turn 1): when did nolan make his batman movies?
Answer: Nolan launched one of the Dark Knight’s most successful eras with Batman Begins in 2005, The Dark Knight in 2008, and the final part of the trilogy The Dark Knight Rises in 2012.

Question (Turn 2): who played the role of alfred?
Answer: ... a returning cast: Michael Caine as Alfred Pennyworth...
This canonical conversation illustrates implicit context in follow-up questions. In turn 2, the role of alfred refers to the one in Nolan’s Batman trilogy; in turn 3, what about refers to the actor playing the role of Harvey Dent (the Batman movies remain as additional context all through turn 5); in turn 5, compared to alludes to the box office reception as the point of comparison. Thus, ConvQA is far more than coreference resolution and question completion [12].

Relevance. Conversational QA lies under the general umbrella of conversational search, that is of notable contemporary interest in the IR community. This is evident through recent forums like the TREC Conversational Assistance Track (CAsT) \(^1\) [7], the Dagstuhl seminar on Conversational Search \(^2\) [1], and the ConvERSe workshop at WSDM 2020 on Conversational Systems for E-Commerce \(^3\). Our proposal CROWN was originally a submission to TREC CAsT, where it achieved above-median performance on the track’s evaluation data and outperformed several neural methods.

Approach and contribution. Motivated by the lack of a substantial volume of training data for this novel task, and the goal of devising a lightweight and efficient system, we developed our unsupervised method CROWN (Conversational passage ranking by Reasoning Over Word Networks) that relies on the flexibility of weighted graph-based models. CROWN first builds a backbone graph referred to as a Word Proximity Network (WPN) that stores word association scores estimated from large passage corpora like MS MARCO \([14]\) or TREC CAR \([8]\). Passages from a baseline model like Indri are then re-ranked according to their similarity weights to question terms (represented as node weights in the WPN), while preferring those passages that contain term pairs deemed significant from the WPN, close by. Such coherence is determined by using edge weights from the WPN. Context is propagated by various models of (decayed) weighting of words from previous turns.

CROWN enables conversational QA over passage corpora in a clean and intuitive UI, with interactive response times. As far as we know, this is the first public and open-source demo for ConvQA over passages. All our material is publicly available at:

- Online demo: https://crown.mpi-inf.mpg.de/
- Walkthrough video: http://qa.mpi-inf.mpg.de/crownvideo.mp4
- Code: https://github.com/magkai/CROWN.

2 METHOD

2.1 Building the Word Proximity Network

Word co-occurrence networks built from large corpora have been widely studied \([2, 9]\), and applied in many areas, like query intent analysis \([16]\). In such networks, nodes are distinct words, and edges represent significant co-occurrences between words in the same sentence. In CROWN, we use proximity within a context window, and not simple co-occurrence: hence the term Word Proximity Network (WPN). The intuition behind the WPN construction is to measure the coherence of a passage w.r.t. a question, where we define coherence by words appearing in close proximity, computed in pairs. We want to limit such word pairs to only those that matter,

\[^1\]\(\text{http://www.treccast.ai/}\)
\[^2\]\(\text{https://bit.ly/2Vf2Qh}\)
\[^3\]\(\text{https://wds-converse.github.io/}\)

\[^2\] \[^3\] Figure 1: Sample conversation and word proximity network.

i.e., have been observed significantly many times in large corpora. This is the information stored in the WPN.

Here, we use NPMI (normalized Pointwise Mutual Information) for word association: \(\text{npmi}(x, y) = \log \frac{\text{p}(x, y) - \text{p}(x) \cdot \text{p}(y)}{\text{p}(x) \cdot \text{p}(y)}\) where \(p(x, y)\) is the joint probability distribution and \(p(x), p(y)\) are the individual unigram distributions of words \(x\) and \(y\) (no stopwords considered). The NPMI value is used as edge weight between the nodes that are similar to conversational query tokens (Sec. 2.2). Node weights measure the similarity between conversational query tokens and WPN nodes appearing in the passage.

Fig. 1 shows the first three turns of our running example, together with the associated fragment (possibly disconnected as irrelevant edges are not shown) from the WPN. Matching colors indicate which of the query words is closest to that in the corresponding passage. For example, Nolan has a direct match in the first turn, giving it a node weight (compared using \text{word2vec} cosine similarity) of 1.0. If this similarity is below a threshold, then the corresponding node will not be considered further (\text{caine} or \text{financial}, greyed out).

Edge weights are shown as edge labels, considered only if they exceed a certain threshold: for instance, the pairs (batman, movie) and (harvey, dent), with NPMI \(\geq 0.7\), qualify here. These edges are highlighted in orange. Edges like (financial, success) with weight above the threshold are not considered as they are irrelevant to the input question (low node weights), even though they appear in the given passages.

2.2 Formulating the Conversational Query

To propagate context, CROWN expands the query at a given turn \(T\) using three possible strategies to form a conversational query \(c_t\). \(c_q\) is constructed from previous query turns \(q_t\) (possibly weighted with \(w_t\)) seen so far:

- **Strategy \(c_q\)**: simply concatenates the current query \(q_T\) and \(q_1\). No weights are used.
- **Strategy \(c_{q2}\)**: concatenates \(q_T, q_{T-1}\) and \(q_1\), where each component has a weight \(w_1 = 1.0, w_T = 0.8, w_{T-1} = (T - 1)/T\).
- **Strategy \(c_{q3}\)**: concatenates all previous turns with decaying weights (\(w_1 = 1.0, w_T = 1.0, w_{T-1} = 0.8\)), as they are usually relevant to the full conversation.
This $cq$ is first passed through Indri to retrieve a set of candidate passages, and then used for re-ranking these candidates (Sec. 2.3).

2.3 Scoring Candidate Passages

The final score of a passage $P_i$ consists of several components that will be described in the following text.

**Estimating similarity.** Similarity is computed using node weights:

$$score_{node}(P_i) = \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{m} C_1(p_{ij}, p_{ik}) \cdot \max_{v \in eq} \left( \frac{\text{sim}(\text{vec}(p_{ij}), \text{vec}(v_{k}))}{w_j} \right)$$

where $C_1(p_{ij})$ is 1 if the condition $C_1(p_{ij})$ is satisfied, else 0 (see below for a definition of $C_1$). $\text{vec}(p_{ij})$ is the word2vec vector of the $j^{th}$ token in the $i^{th}$ passage; $\text{vec}(v_{k})$ is the corresponding vector of the $k^{th}$ token in the conversational query $cq$ and $w_j$ is the weight of the turn in which the $k^{th}$ token appeared; $\text{sim}$ denotes the cosine similarity between the passage token and the query token embeddings. $C_1(p_{ij})$ is defined as $C_1(p_{ij}) := \exists v_{k} \in eq : \text{sim}(\text{vec}(p_{ij}), \text{vec}(v_{k})) > \alpha$ which means that condition $C_1$ is only fulfilled if the similarity between a query and a passage word is above a threshold $\alpha$.

**Estimating coherence.** Coherence is calculated using edge weights:

$$score_{edge}(P_i) = \sum_{j=1}^{n} \sum_{k=1}^{m} C_2(p_{ij}, p_{ik}) \cdot \text{NPMI}(p_{ij}, p_{ik})$$

where $C_2(p_{ij}, p_{ik}) := \text{hasEdge}(p_{ij}, p_{ik}) \land \text{NPMI}(p_{ij}, p_{ik}) > \beta$

Estimating positions. Passages with relevant sentences earlier should be preferred. The position score of a passage is defined as:

$$score_{pos}(P_i) = \max_{s_j \in P_i} \left( \frac{1}{2} \left( \text{score}_{node}(P_i)[s_j] + \text{score}_{edge}(P_i)[s_j] \right) \right)$$

where $s_j$ is the $j^{th}$ sentence in passage $P_i$ and $\text{score}_{node}(P_i)[s_j]$ is node score for the sentence $s_j$ in $P_i$.

Estimating priors. We also consider the original ranking from Indri, which can often be very useful: $score_{indri}(P_i) = 1/\text{rank}(P_i)$ where $\text{rank}$ is the rank that the passage $P_i$ received from Indri.

Putting it together. The final score for a passage $P_i$ consists of a weighted sum of these four individual scores: $score(P_i) = h_1 \cdot score_{indri}(P_i) + h_2 \cdot score_{node}(P_i) + h_3 \cdot score_{edge}(P_i) + h_4 \cdot score_{pos}(P_i)$, where $h_1, h_2, h_3$ and $h_4$ are hyperparameters tuned on TREC CAsT data. The detailed method and the evaluation results of CROWN are available in our TREC report [11]. General information about CAst can be found in the TREC overview report [7].

3 SYSTEM OVERVIEW

An overview of our system architecture is shown in Fig. 2. The demo consists of a frontend and a backend, connected via a RESTful API.

**Frontend.** The frontend has been created using the Javascript library React. There are four main panels: the search panel, the panel containing the sample conversation, the results panel, and the advanced options panel. Once the user presses the answer button, their current question, along with the conversation history accumulated so far, and the set of parameters, are sent to the backend. A detailed walkthrough of the UI will be presented in Sec. 4.

**Backend.** The answering request is sent via JSON to a Python Flask App, which works in a multi-threaded way to be able to serve multiple users. It forwards the request to a new CROWN instance which computes the results as described in Sec. 2. The Flask App sends the result back to the frontend via JSON, where it is displayed on the results panel.

Implementation Details. The demo requires ~170 GB disk space and ~20 GB memory. The frontend is in Javascript, and the backend is in Python. We used pre-trained word2vec embeddings that were obtained via the Python library gensim4. The Python library spaCy5 has been used for tokenization and stopword removal. As previously mentioned, Indri6 has been used for candidate passage retrieval. For graph processing, we used the Python library NetworkX7.

4 DEMO WALKTHROUGH

Answering questions. We will guide the reader through our demo using our running example conversation from Sec. 1 (Fig. 3).

---

4 https://radimrehurek.com/gensim/
5 https://spacy.io/
6 https://www.lermurproject.org/indri.php
7 https://networkx.github.io/
The demo is available at https://crown.mpi-inf.mpg.de. One can start by typing a new question into the search bar and pressing Answer, or by clicking Answer Sample for quickly getting the system responses for the running example.

When did Nolan make his Batman movies?

![Results for Turn 1: when did nolan make his batman movies?](image)

**Figure 4:** Search bar and rank-1 answer snippet at turn 1.

Fig. 4 shows an excerpt from the top-ranked passage for this first question (when did Nolan make his Batman movies?), that clearly satisfies the information need posed in this turn. For quick navigation to pertinent parts of large passages, we highlight up to three sentences from the passage (number determined by passage length) that have the highest relevance (again, a combination of similarity, coherence, position) to the conversational query. In addition, important keywords are in **bold**: these are the top-scoring nodes from the WPN at this turn.

The search results are displayed in the answer panel below the search bar. In the default setting, the top-3 passages for a query are displayed. Let us go ahead and type the next question (who played the role of Alfred?) into the input box, and explore the results (Fig. 5).

Again, we find that the relevant nugget of information (...Michael Caine as Alfred Pennyworth...) is present in the very first passage. We can understand the implicit context in ConvQA from this turn, as the user does not need to specify that the role sought after is from Nolan’s Batman movies. The top nodes and edges from the WPN are shown just after the passage id from the corpus: nodes like batman and nolan, and edges like (batman, role). These contribute to interpretability of the system by the end-user, and help in debugging for the developer. We now move on to the third turn: and what about Harvey Dent?, as shown in Fig. 6. Here, the context is even more implicit, and the complete intent of role in Nolan’s Batman movies is left unspecified. The answer is located at rank three now (see video).

**Figure 6:** Answer at the 3rd-ranked passage for turn 3.

Similarly, we can proceed with the next two turns. The result for the current question is always shown on top, while answers for previous turns do not get replaced but are shifted further down for easy reference. In this way, a stream of (question, answer) passages is created. Passages are displayed along with their id and the top nodes and edges found by CROWN. In the example from Figure 5 not only alfred and role but also batman and nolan, which have been mentioned in the previous turn, are among the top nodes.

**Clearing the buffer.** If users now want to initiate a new conversation, they can press the Clear All button. This will remove all displayed answers and clear the conversation history. In case users just want to delete their previous question (and the response), they can use the Clear Last button. This is especially helpful when exploring the effect of the configurable parameters on responses at a given turn.

**Figure 7:** Advanced options for an expert user.

**Advanced options.** An expert user can change several CROWN parameters, as illustrated in Fig. 7. The first two are straightforward: the number of top passages to display, and to fetch from the underlying Indri model. The node weight threshold $\alpha$ (Sec. 2.3) can be tuned...
CROWN is an unsupervised approach for conversational passage ranking. Answers are retrieved from MS MARCO and TREC CAR datasets. We formulated the objective of maximizing the passage score for a query as a combination of similarity and coherence. Passages are preferred that contain words semantically similar to the words used in the question. Coherence is expressed using term proximity: We built a word-proximity network from the corpus, where words are nodes and there is an edge between two nodes if they co-occur in the same passage in a statistically significant way, within a context window. We use NPMI (normalized pointwise mutual information) as a measure of this word association significance.

Our code and further technical information are available here. For a complete demo walkthrough, have a look at our video.

Conversational Question Answering over Passages by Leveraging Word Proximity Networks

For feedback and clarifications, please contact:

Magdanela Kaiser (email: AT MPI INF DOT mpg DOT de)
Rishiraj Saha Roy (email: AT MPI INF DOT mpg DOT de)
Gerhard Weikum (weikum AT mpi INF DOT mpg DOT de)

To know more about our research group on question answering, please visit https://hpi-aisig DOT mpg.

Figure 8: A summarizing description of CROWN.

We demonstrated CROWN, one of the first prototypes for unsupervised conversational question answering over text passages. CROWN resolves implicit context in follow-up questions by expanding the current query with keywords from previous turns, and uses this new conversational query for scoring passages using a weighted combination of similarity, coherence, and positions of approximate matches of query terms. In terms of empirical performance, CROWN scored above the median in the Conversational Assistance Track at TREC 2019, being comparable to several neural methods. The presented demo is lightweight and efficient, as evident in its interactive response rates. The clean UI design makes it easily accessible for first-time users, but contains enough configurable parameters so that experts can tune CROWN to their own setups.

A very promising extension is to incorporate answer passages as additional context to expand follow-up questions, as users often formulate their next questions by picking up cues from the responses shown to them. Future work would also incorporate fine-tuned BERT embeddings and corpora with more information coverage.

5 CONCLUSION

We demonstrated CROWN, one of the first prototypes for unsupervised conversational question answering over text passages. CROWN resolves implicit context in follow-up questions by expanding the current query with keywords from previous turns, and uses this new conversational query for scoring passages using a weighted combination of similarity, coherence, and positions of approximate matches of query terms. In terms of empirical performance, CROWN scored above the median in the Conversational Assistance Track at TREC 2019, being comparable to several neural methods. The presented demo is lightweight and efficient, as evident in its interactive response rates. The clean UI design makes it easily accessible for first-time users, but contains enough configurable parameters so that experts can tune CROWN to their own setups.

A very promising extension is to incorporate answer passages as additional context to expand follow-up questions, as users often formulate their next questions by picking up cues from the responses shown to them. Future work would also incorporate fine-tuned BERT embeddings and corpora with more information coverage.

REFERENCES