NtoN: end-to-end dialog system

Lukas Zilka

1 Database

Conceptually a database is for us a static mapping from queries to results, where both queries and results are elements from a vocabulary V of size $W = |V|$. The mapping works in the following way: Query is matched to a collection of database entries (that define the database) and the most matching entry is found and used to generate the result. The collection of database entries is a set $E = \{e_1, ..., e_K\}$, where $e_i = \langle query, answer \rangle$.

To make such database useful in gradient-based algorithms we need to make the result retrieval mechanism differentiable.

For this purpose we define a differentiable function $f(x)$, which transforms a query x to a result $f(x)$. Both x and $f(x)$ are distributions over words. For easier understanding it is better to assume x is a 1-hot encoded word from V.

Definition:

$$f : \mathbf{R}^W \to \mathbf{R}^W$$

$$f(x) = softmax(\langle r_1(x), ..., r_W(x) \rangle)$$

where *result* $r_w(x)$ intuitively expresses how much does given query x result in word w in the output (database result):

$$r_w(x) = \sum_{k=1}^{K} \langle R_w, m(x) \rangle$$

and *match* $m(x)$ expresses how much does given query x match entries in database:

$$m(x) = \langle m_1(x), ..., m_K(x) \rangle$$

$$m_k(x) = \langle M_k, x \rangle$$

Matrix M defines which query words match given database entry, and matrix R defines what output words does given database entry belong to. They together encode the collection of database entries E.

2 Extensions

2.1 Multi-turn

2.2 More queries to database in 1 turn

2.3 Real Data

2.4 Assembling query from more tokens

2.5 Larger database