Intent Features for Rich Natural Language Understanding

Brian Lester[♠*], **Sagnik Ray Choudhury**[◇†], **Rashmi Prasad**[♠], **Srinivas Bangalore**[♠]
[♠]Interactions, 41 Sprint Street, New Providence, NJ 07974
[◇]University of Copenhagen, Denmark
`{blester, rprasad, sbangalore} @interactions.com,`
`src@di.ku.dk`

Introduction

Nuance and Intent Space :
In a production dialogue system, there are often similar situations that require drastically different responses. For example, "I want to cancel my subscription." and "I am thinking about canceling my subscription." are very similar. They are both about the canceling of a subscription. However, they differ in the users conviction. The latter user is much more likely to not cancel if offered a discount. Making this distinction is critical for creating sophisticated and nuanced dialogue systems. A common approach to solve this problem would be to split the intent space so the dialogue manager can differentiate between these examples, creating a `cancel` and a `think-cancel` intent. Using intents to recognize specific situations leads to data sparsity as each intent is broken into many sub-categories like present vs. past tense, how certain a user is in their actions, and if the user has tried an action or not. There would be very few examples of each intent. Additionally, the combinations of different sub-categories would cause a combinatorial explosion of intents. Another short-coming of fine-grained intents is the loss of compositionality. Fundamentally the `cancel` and `think-cancel` intents are very similar, but because they are modeled as independent output classes, there is not a shared representation of these labels the model can lean on.

Intent Features

Communicative functions: The communicative functions (cf) captures what kind of response (or action) the user is trying to elicit from the system. We define five such functions:

- `inform`: The user is informing the system about something. Typically, these intents are a response to a question or they represent background information surrounding the main purpose of the utterance. For example, in the utterance, "I am installing X but it keeps saying I have an error", the first clause has a communicative function of `inform`. The user provides background information about installing something on a device and then presents a problem with the install procedure, which would have a communicative function of `issue`.
- `issue`: The user is saying that something has gone against their expectations (see above for an example).
- `request-action`: The user requests for some action to be undertaken in response to the request, or requests help with something. For example, "I would like to install X."
- `request-confirm`: The user is requesting confirmation, or disconfirmation, of their belief. Often this warrants a yes/no answer. For example, one expects a yes or no from, "Was my installation successful?"
- `request-info`: The user is requesting some information about something. These are typically expressed as "wh/how" questions, such as: "How can I install X?"

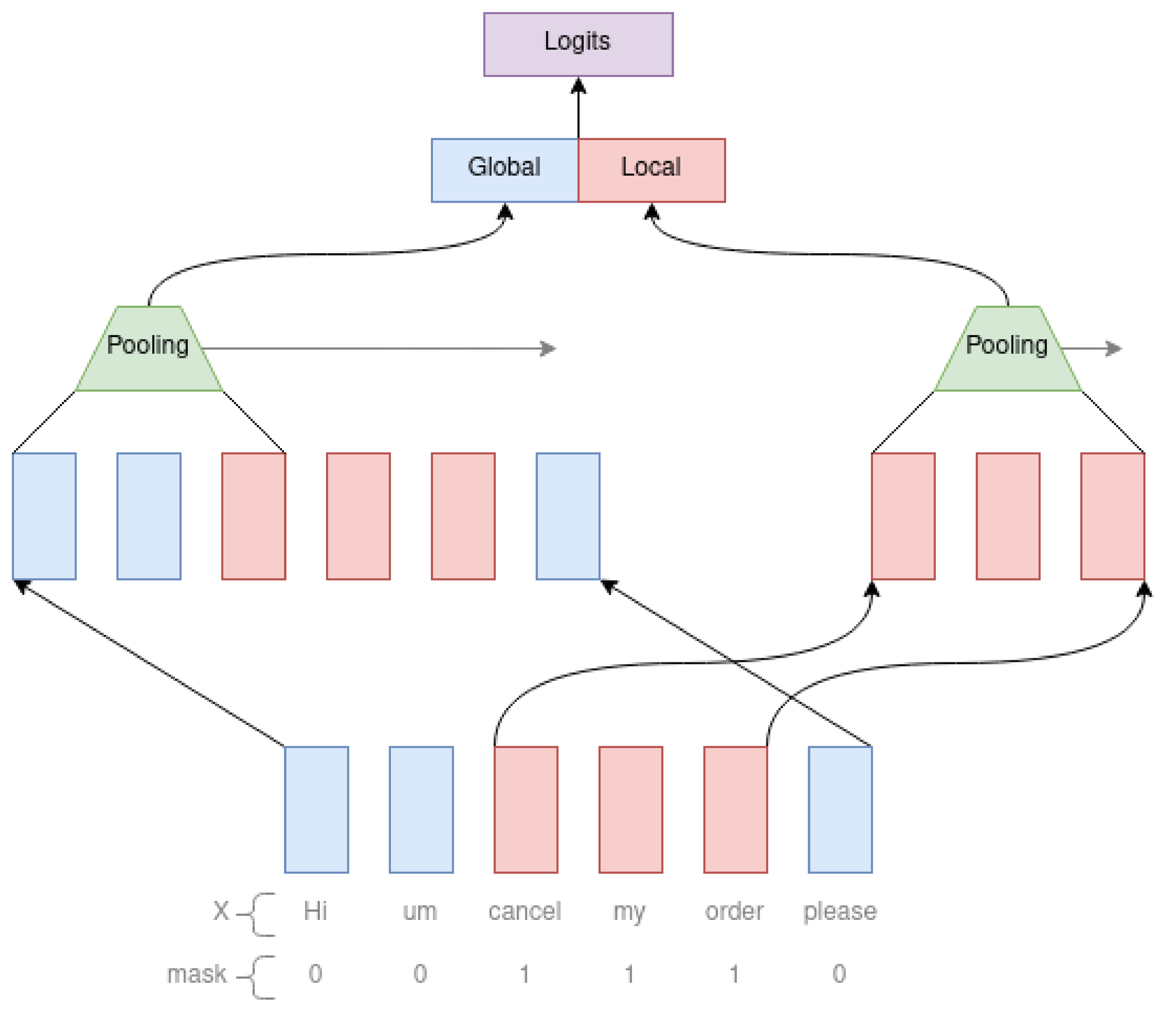
Attribution: Attribution is concerned with *agency*. There are two types of attribution. The first type is the of attribution of the communicative function (**attr-cf**) and it deals with who is the primary source of the content of the topic. The second type is the attribution of the event/action (**attr-ev**) of a topic and describes who is the agent of the event or action. Both **attr-cf** and **attr-ev** take values `self` (when the agent is the user) and `other`.

Negation: Topics of many intents are represented in their negated versions, as well. For example, in the software domain, the `compatibility` intent models whether a piece of software is compatible with some device. A negation feature would denote incompatibility. The negation feature takes values `positive` and `negative`.

Tense: Events and actions can occur in the past, present, or future, which is modeled by the tense feature using values of `past`, `present`, or `future`. The steps to solve a problem as it occurs are often quick-fixes, whereas the first step when fixing a problem that occurred in the past is often information gathering. The tense feature allows the dialogue manager to distinguish between these two possibilities.

Modality: The real-world actions and events represented by an intent can also be viewed in terms of a modality of certainty, that is, whether or not the event or action actually occurred, and to what degree. We consider two types of modality. The first is *possibility*—the expression of the event as hypothetical, or being possible, rather than certain, as in, "I am *planning/going* to install X on my laptop." We also consider *attempts at action*. An expression can imply that it is unclear whether the action was completed or is in the attempted stage. This is expressed with modifying verbs, such as, "try", as in, "I am *trying* to install X." This feature takes the values `modal-poss`, `modal-try`, and `other`.

Global Local Model [‡]



This model aims to create a targeted representation for a subsection of an utterance while also infusing information derived from the whole utterance. An utterance U of n tokens and a subsequence of k tokens from U, are first encoded into matrices of dimension n x e and k x e, respectively, where e is the dimension of some shared embedding space. A "global" pooling function $g : \mathbb{R}^{n \times e} \mapsto \mathbb{R}^{e}$ collapses the global sentence matrix to a sentence vector and another "local" pooling function $l : \mathbb{R}^{k \times e} \mapsto \mathbb{R}^{e}$ reduces the span matrix to a span vector. These vectors are concatenated to create the final representation for the span S. This representation is then projected into the output space.

Results

Feature	BiLSTM-CRF Cascaded Tagger	Span-level Convolutional	Global-Local
Attribution CF	91.90	95.37	**97.69**
Attribution EV	92.27	95.86	**98.16**
Communicative Function	89.22	90.12	**91.92**
Modality	92.61	96.60	**99.36**
Tense	86.01	89.31	**92.59**
Negation	94.45	95.86	**98.73**

The BiLSTM tagger has the worse performance, even When intent information is supplied. The span-level convolutional model, which is agnostic to the tokens of the other spans, performs much worse than the Global-Local model, which shows that global utterance information is valuable.

Ablations

Model	Attr-cf	Attr-ev	CF	Modality	Tense	Negation
Global-Local	**97.69**	**98.16**	**91.91**	**99.36**	**92.59**	**98.73**
– Global Context	93.55	95.47	90.14	96.74	87.18	95.74
– Shared Embedding	97.65	96.63	91.43	98.34	90.17	96.36

We ablate the Global-Local model's increased parameter count by using *only* the span as the input (as opposed to *both* the utterance and the span). As we can see in the "– Global Context" row, limiting the model to only see the span causes large performance drops across the board. This model is even worse than the simple convolutional model. This implies that the global context is critical.

Our ablations in the "– Shared Embedding" row shows that using a shared embedding space does yield performance gains, but it can be removed for the sake of easier model deployment and still maintain superior performance over the span-level model.

Conclusion

We define intent features, a core set of general annotations, on intents that provide context and clarity on the exact nature of the user requests, and allow for a more natural and intelligent response from the dialogue manager. A NLU system that produces these intent features has been deployed in a production system with a dialogue manager that makes use of them.

We propose a new neural network architecture, the Global-Local model, that fuses the representation of the content of a span of text and its global context through learned pooling functions. This model shows large improvements over several strong baselines.

[†]Now an AI Resident at Google
[†]Work done while at Interactions
[‡]https://github.com/blester125/global-local-model