Feature Structures in the Wild: A Case Study in Mixing Traditional Linguistic Knowledge Representation with Neural Language Models

Gerald Penn and Ken Shi
Department of Computer Science
University of Toronto
{gpenn,kenshi}@cs.toronto.edu

Abstract

This paper briefly presents an evaluation of three models: a domain-specific one based upon typed feature structures, a neural language model, and a mixture of the two, on an unseen but in-domain corpus of user queries in the context of a dialogue classification task. We find that the mixture performs the best, which opens the door to a potentially new application of neural language models. A further examination of the domain- We also consider the inner workings of the domain-specific model in more detail, as well as how it came into being, from an ethnographic perspective. This has changed our perspective on the potential role of structured representations in the future of dialogue systems, and suggests that formal research in this area may have a new role to play in validating and coordinating *ad hoc* dialogue systems development.

1 Introduction

While contemporary NLP research marvels at how closely a simple neural language model can come to a coherent conversation partner in dialogue tasks, it nevertheless remains true that language models, neural or otherwise, are difficult to adapt in a manner that keeps both the responses constructive and the number of dialogue turns to a minimum in settings where users expect a rapid and successful conclusion to their information-seeking interactions.

Over the past year, we have worked with an industrial partner, iNAGO, Inc., a specialist in conversational agents in domains such as product information and control, navigation and automotive driver assistance, to find ways in which recent developments in dialogue systems could improve their products. Focussing on dialogue act classification at the outset, we did indeed find a way to make a simple but important improvement, which is described below, but what struck us as equally salient is just how well their system works to begin with, relative to research systems currently in circulation.

A subsequent investigation of just how their system works has revealed some novel simplifications of concepts that should be very familiar territory to this audience: typed feature structures, user modelling and semantic distances defined through a combination of lattice-theoretic calculations on epistemic networks and similarity coefficients. The novelty arises to a great extent because the developers at iNAGO were mostly unfamiliar with research publications on these topics, and so the resemblance of their proposed solution to a variety of representational strategies that have been used in the dialogue research community over the last 30 years is in itself noteworthy.

But the reason that our improvement works, we believe, stems from the complementarity of this dialogue classifier and the language-modelling-based approach that we combined it with. This complementarity needs to be investigated in more detail in a wider range of domains and across languages with a wider distribution of resource availabilities (we have experimented only with English-language systems), but it opens the door to a possible application of neural language models that has hitherto not been considered, possibly because of misbegotten claims of their cognitive plausibility, which are in turn more suggestive of their use exclusively as drop-in replacements. Domain-specific models are known to have problems with coverage, particularly outside their domains. Large-scale neural language models do not have this problem, even if their within-domain performance is somewhat lackluster by comparison. Even the simple combination of the two that we tried appears to address the weaknesses of both of these approaches in isolation.

Proceedings of the ESSLLI 2021 Workshop on Computing Semantics with Types, Frames and Related Structures, pages 53–59
Utrecht, The Netherlands (online), 26–27 July, 2021.

We will begin with a discussion of the general approach to mixing the results of these two approaches to dialogue act classification, and then return to how iNAGO's system computes its own results.

2 Re-ranking

Re-ranking is a simple method for combining discriminative and generative models that takes the top answers from the generative model, in order of preference, and merely changes the order of preference using information from the discriminative model. The top answers from the generative model, which serve as the inputs to the re-ranker, are often known as *candidates*. In our case, these candidates are generated by iNAGO's system in response to a user-provided query. It should be noted that in this particular application, the user-provided query is also made available to the discriminative model, which is not always the case in re-ranking.

3 Task and Models

We evaluated three models: iNAGO's classifier, without re-ranking, the responses of a BERT-based dialogue act classifier, and the result of mixing the two models, which we shall refer to as the *Mixed model*.

With all three models, the task is to classify the transcription of a query spoken by an automobile driver according to several hundred predetermined classes of query that the system is capable of answering, based upon information about the vehicle, the state of the vehicle at the time of the query, and other information from map resources, etc., as needed. The result is a list of classes, in decreasing order of their confidence scores. Higher confidence answers have lower ordinate rank, i.e., the best answer, of rank 1, is the class with the highest confidence. The presumption is that the answer corresponding to the class with the highest confidence in the database of predetermined classes would be returned to the user when this model is used.

Our mixture method crucially relies upon the availability of these confidence scores.

The BERT-based classifier uses the pre-trained model distributed with the original paper (Devlin et al., 2019), and adds the three levels of embeddings (Figure 1) into a single vector that represents an entire string of input. Queries are classified by computing the cosine similarity of the vector for the query with the vectors for each of a list of sentences, one for each class in the database, that characterizes a prototypical question for that class, very much as an FAQ list would. Again the classes are ranked by this similarity score.

The Mixed model takes evidence from both iNAGO's model and the BERT model into consideration. It does so by treating iNAGO's confidence score, c_i, as a mixture parameter, and computes the sequence:

$$m_i = c_i \cdot a_i + (1 - c_i) \cdot b_i$$

for each candidate class, where a_i is the rank assigned by iNAGO's model, and b_i is the rank assigned by the BERT model. The new ranking of the candidates is then given by sorting the candidates in decreasing order of m_i.

4 Data

The queries that were used in our experiments were automatically generated using the method of Zheng (forthcoming) from documentation provided by a Tier-1 auto manufacturer. The documentation was only provided in April, 2021, and were thus not available either to the present authors or to iNAGO during the development of its model. The corpus consists of 232 queries.

Ground-truth answers were not made available for any of the queries by the manufacturer, but iNAGO manually mapped the corresponding answers generated by Zheng (forthcoming) to the most suitable class within their database of prototype questions and answers. iNAGO provided us with their model's rankings and confidence scores, as well as a complete list of the 410 classes and prototypes from the database that their model could refer to. As a result, we are capable of computing scores that evaluate these models in a dialogue turn classification task, but not overall measures of dialogue quality such as number of turns to completion, or the percentage of accomplishment of a stated user goal.

Among the 232 queries, 20 of them were *unrecalled*, meaning that an appropriate class was available within iNAGO's database (it was for all 232), but was not in iNAGO's model's candidate list. The existence of these instances precludes the use of an average precision score directly, as is standard in query-reranking approaches to internet search or information retrieval systems, on any

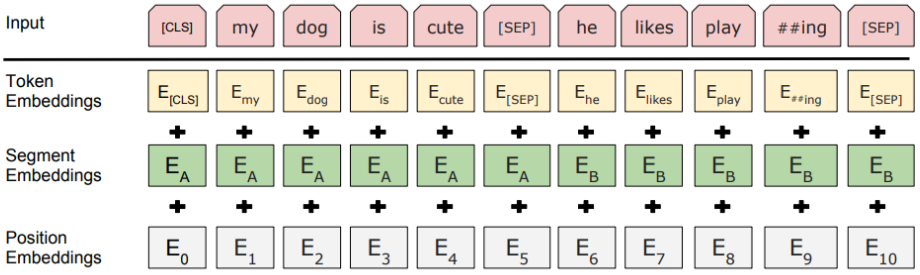

Figure 1: A graphical depiction of the computation of BERT embeddings (Devlin et al., 2019).

of our three models except BERT, which always produces a score.

Below, we report our evaluation of the three models on two query samples from this dataset: *Without Unrecalled Cases*, in which only the 212 queries for which the correct class label was recalled at any rank are used, and *With Unrecalled Cases*, which considers all 232. For the latter sample, for the purposes of computing the Mixed model's ranking of the 20 unrecalled queries, the manually annotated class is appended at the bottom of the list of candidate classes for iNAGO's model, with a confidence of zero.

5 Evaluation Scores

We used three different scores to evaluate each model. All can be regarded as derivative measures of performance, although they have applications to further exploratory data analysis, such as through visualization.

We compute the *mean*, *variance* and *median* of the rank of the ground truth category in each candidate list. In the case where confidences can be interpreted as probabilities, this corresponds to a data likelihood score.

We also compute the *mean reciprocal rank (MRR)*, which is a variant of mean average precision. The formula of MRR is:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}$$

where N is either 212 or 232 (see Section 4) and $rank_i$ denotes the rank of item i in a list. MRR is a classification accuracy measure that bestows partial credit for answers of rank greater than 1, according to a hyperbolic curve.

Finally, we compute the *top-1 accuracy* of the model. Here, we simply consider the percentage of cases where the model assigned the top rank (1) to the manually annotated class.

6 Model Evaluations

The evaluation scores are given in Tables 1–2. See also Figures 2–4 for the counts of the manually annotated label's rank (the highest was 75) without consideration of unrecalled cases, and Figures 5–8 for counts including unrecalled cases. The generally hyperbolic shape of those distributions compels us to compute the logarithms of the counts at each rank and fit those logarithms to a line with slope B using least-squares regression, having coefficient of determination, R^2.

7 Discussion of Results

There are two points that can be clearly ascertained. The first is that iNAGO's model is to be credited for its generally better performance on these queries, which were unseen during the development of that model, but mostly in-domain. BERT is widely regarded as not an easy model to beat, and iNAGO's model did beat it soundly in both conditions across all measures. As the histograms show, iNAGO's model and the Mixed model are also both generally sharper around the top rank than BERT.

That a linear combination of two independent, unbiased estimators should exist that lowers variance is to be expected. On the other hand, we did not determine the mixture parameter by directly optimizing variance or covariance. Note also that the iNAGO model's variance was already low, when it was able to locate the correct class label at any rank. This suggests that the mixed model's improvement to the iNAGO model was primarily through its greater coverage.

The second point is that iNAGO's and BERT's performances are complementary enough to be of

	Mean (Var)	Median	MRR	T1	$-B$	R^2
iNAGO	1.594 (2.489)	**1.0**	**0.854**	**77.83%**	0.866	0.741
BERT	3.675 (16.145)	2.0	0.575	41.04	0.187	0.701
Mixed	**1.552** (2.903)	**1.0**	0.851	75.00	**1.031**	0.933

Table 1: Performance on Dialogue Classification Task, not including unrecalled cases.

	Mean (Var)	Median	MRR	T1	$-B$	R^2
iNAGO	2.263 (10.766)	**1.0**	0.798	73.28%	0.266	0.407
BERT	3.560 (15.382)	2.0	0.590	43.10	0.259	0.567
Mixed	**1.629** (3.239)	**1.0**	**0.841**	**74.14**	**1.544**	0.835
BERT (unrecalled cases only)	2.350 (6.029)	1.0	0.748	65.00	0.360	0.481

Table 2: Performance on Dialogue Classification Task, including unrecalled cases.

mutual benefit to each other. This is particularly true when we consider the cases unrecalled by iNAGO's model on their own, where BERT's performance is so good that the Mixed model's performance on all 232 cases has a mean rank of less than 2.

With a corpus of this size, fine-tuning BERT was beside the point, and so this experiment was conducted in a zero-shot setting. On the other hand, the BERT model that assigned ranks within the mixed model was also the base model.[1] Corpora of this size are not uncommon to dialogue system designers, and so this is an ecologically valid setting.

8 How Did They Do It?

The better performance of iNAGO's model naturally compelled us to ask how it works. The answer is surprising in just how unsurprising it is. It begins with a round of slot/filler labelling inside the query or candidate prototype using a sequential labeller, very much as one finds in the ATIS NLU task (Niu and Penn, 2019). Three linear passes over the annotated string with very small cascades of between one and three rules lead to the iterative construction of a data structure that is essentially identical to a typed feature structure (Carpenter, 1992). The signature of the formalism contains 17 features and a type hierarchy consisting of around 40 000 types, although all but about 4 000 of those types are proper nouns that designate types of cuisine, landmarks, titles of songs, etc. The feature structures represent a combination of propositional content and user intention, the latter being classifiable into 11 discrete types.

8.1 Rules

The first cascade of rules looks only for evidence that the input is or is not a continuation of an earlier dialogue, and then classifies the input by user intention. The second cascade fills in or refines the value types of features that have been determined to exist either (1) by the intention type, (2) by the presence of a particular slot filler in the labelled input sequence or (3) by previous dialogue turns in the case of a continuation. The "filling" is monotonic and is consistent with the type-inferencing rules of Carpenter (1992) that are used to compute what he terms most general satisfiers of expressions from a Rounds-Kasper-style attributed description language.

The third cascade modifies the feature structure non-monotonically if it matches a template defined by one of its rules. This stage essentially handles exceptions that could not be accommodated by the second stage. Each rule in this cascade handles one exception each. Templates can detect:

1. whether the value at a feature path has been refined by the second cascade as a result of the current input,
2. whether a feature value bears a subtype of a given type,
3. whether a feature value is exactly a given type, and
4. whether the value of one of a finite number of extra-logical variables is equal to a given constant,

[1] An anonymous reviewer suggested that we attempt to fine-tune the BERT model on this dataset, in spite of obvious concerns about the generalization bound on a set of this size. Indeed, performance was worse with respect to every measure after fine-tuning.

5. closed under conjunction and disjunction.

The extra-logical variables are set by the state of the automobile. Impressively, however, there are only two rules/exceptions in the third cascade.

8.2 Similarity

Given a pair of these feature structures, one for a query and one for a candidate, their similarity is determined through a modified form of a weighted Jaccard index acting upon a set-theoretic reduction of the two structures. In this reduction, both feature structures are reduced to sets of feature paths terminating in a value that consists of a type and no other substructures. The actual lengths of the feature paths are irrelevant to the similarity score, but serve to identify like values between the two feature structures that can be compared. Given $|K|$ such paths, on which at least one of feature structures F and G define a value, let us call F_k (resp. G_k) the value of F (resp. G) on path $k \in K$, where it is defined, and the most general type, $\bot$ (in the orientation of Carpenter (1992) — many others would call it $\top$), elsewhere.

While each value is merely a type with no appropriate features, that type is situated within a type hierarchy. This graph of subtyping relations is assumed by Carpenter (1992) to be a meet semilattice for convenience, as it is here. Let $h(\tau)$ be the height of type τ, where the height of a type is taken to be the length of the longest chain from $\bot$ to that type. The A-similarity of F and G is then definable as:

$$A(F,G) = \frac{\Sigma_k w_k \cdot h(F_k \sqcap G_k)}{\Sigma_k w_k \cdot \max(h(F_k), h(G_k))},$$

where $\sigma \sqcap \tau$ is the meet of types σ and τ. It is this A-similarity that is returned as iNAGO's confidence score. Note that its range is $[0,1]$ when $\Sigma_k w_k = 1$ and that high values are attained through a combination of (1) there being many (vs. few) paths on which both the query and a candidate have values defined, and (2) those values having very high (vs. low) meets. The meets are maximally high when both $F_k = G_k$ and F_k takes on a very high/specific value. iNAGO determined the weights w_k through *ad hoc* experimentation on labelled training queries that had been obtained from a different source.

The use of the height of a meet, or the depth of a least common superconcept (LCS) in the parlance of lexical semanticists, dates back to the *conceptual similarity* score of Wu and Palmer (1994), although there it is used as a normalizer on the length of the walk from F_k to G_k via their LCS in the semilattice. The walk lengths from either F_k or G_k to $F_k \sqcap G_k$ are not taken into account in A-similarity.

The iNAGO model ranks the prototypes of its classes by their A-similarity to each query, subject to two thresholds. First, no more than the top 75 classes can be returned. Second, no class with an A-similarity of less than 0.1 can be returned. These thresholds were set empirically. Only one query returned a list that was truncated at 75. The median length of ranked class labels was 11.

8.3 Nomenclature

It is clear from the nomenclature used in company-internal documentation that the developers of this system had not read Carpenter (1992), nor anything else about typed feature logic, or feature-based grammar development. Types are referred to as "entities," features as "roles," feature paths as "criteria," typed feature structures as "interpretations," the type hierarchy as a "criteria set," and chains of types in the hierarchy as "paths." Their knowledge of these structured representational devices seems to factor exclusively through the same early-1980s research on programming language theory and inclusional polymorphism that was so influential on both typed feature logic and linguistic formalisms such as HPSG (particularly the earlier, pre-1994 versions of it; Pollard and Sag, 1987), on the one hand, and modern, object-oriented, imperative programming language constructs, on the other.

As a result, we see no evidence for any sort of deeper convergence or objective suitability of the formalism for dialogue analysis that iNAGO happened upon. Instead, we claim that this manner of structured representation had become, and arguably remains, the *de facto* strategy for reasoning about language and dialogue among university-educated software engineers. The real question may therefore be not how they did it, but why they would ever have done anything else.

9 Conclusion

This paper briefly presented an evaluation of three models, a domain-specific one based upon typed feature structures, a neural language model and a mixture of the two, on an unseen but in-domain corpus of user queries. Our first recommendation is therefore that mixtures of semanti-

cally rich, conventional dialogue classifiers with neural language models should be investigated further, as our results suggest that they can produce the best combination of classifier accuracies and coverage.

We then considered the domain-specific model in more detail. While it is probable that the approach taken by this model would not scale up well to very large domains on its own, to say nothing of domain-independent dialogue modelling, it is indeed difficult to fathom why this manner of reasoning about dialogue should simply go away. Software developers, it appears, need no particular formal instruction in order to create them, perhaps apart from some standardization of their terminologies. Domain-specific approaches very apparently can still achieve higher levels of performance than what black-box semantic embeddings are currently capable of.

Our second recommendation is therefore the same as our first recommendation: we *really* should, as a community, encourage this sort of model combination as a means of enabling and enhancing what software developers are already doing without our permission. It not only improves the accuracy of the systems they are building, but may provide a low-cost means of relaxing domain restrictions. Our third recommendation is that those engaged in the formal study of structured representations should develop their unique capacity to provide the means for validating and coordinating domain-specific dialogue systems that spring up "in the wild," which will allow us to harness a very large pool of unspecialized talent to advance the state of the art in this field.

References

B. Carpenter. 1992. *The Logic of Typed Feature Structures: With Applications to Unification Grammars, Logic Programs and Constraint Resolution*. Cambridge Tracts in Theoretical Computer Science. CUP.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL*, pages 4171–4186.

J. Niu and G. Penn. 2019. Rationally reappraising ATIS-based dialogue systems. In *Proc. 57th ACL*, pages 5503–5507.

C. Pollard and I. Sag. 1987. *Information-based Syntax and Semantics*. Number 13 in CSLI Lecture Notes. CSLI Publications.

Z. Wu and M. Palmer. 1994. Verb semantics and lexical selection. In *Proc. 32nd ACL*, pages 133–138.

Chenxing Zheng. forthcoming. Self-guided question generation with transformers. Master's thesis, York University.

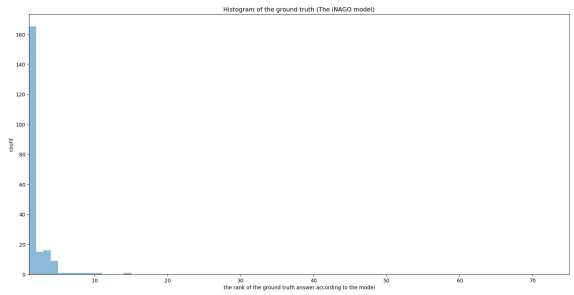

Figure 2: Count of ground truth's rank (iNAGO's model), without unrecalled cases.

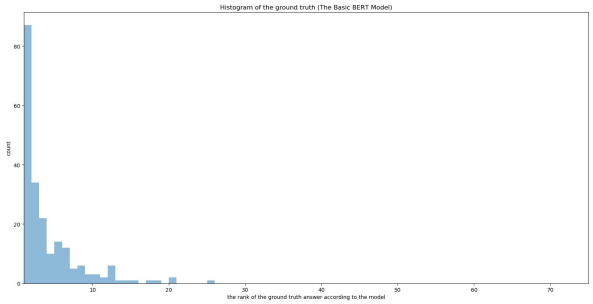

Figure 3: Count of ground truth's rank (BERT model), without unrecalled cases.

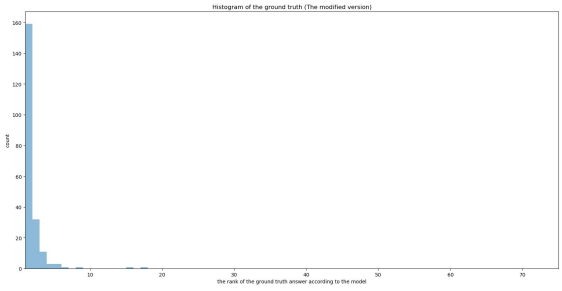

Figure 4: Count of ground truth's rank (Mixed model), without unrecalled cases.

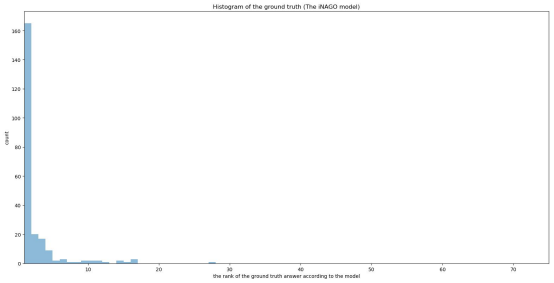

Figure 5: Count of ground truth's rank (iNAGO's model), with unrecalled cases.

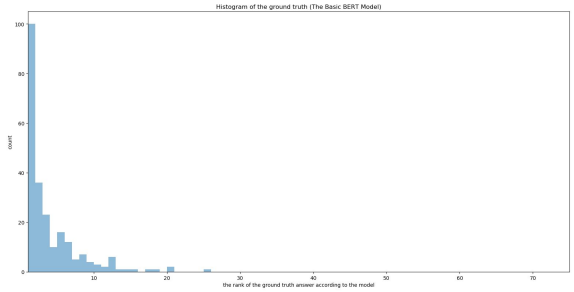

Figure 6: Count of ground truth's rank (BERT model), with unrecalled cases.

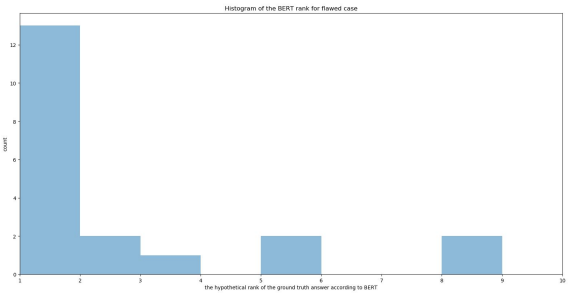

Figure 7: Count of ground truth's rank (BERT model), unrecalled cases only.

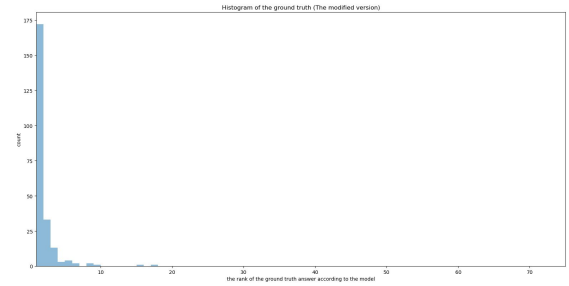

Figure 8: Count of ground truth's rank (Mixed model), with unrecalled cases.