André Martins Following
VP of AI Research at Unbabel and Invited Professor at the University of Lisbon.
Aug 17 · 16 min read

ICML+ACL'18: Structure Back in Play, Translation Wants More Context

A few weeks ago I attended the International Conference in Machine Learning (ICML 2018) in Stockholm and, right after, the Annual Conference of the Association for Computational Linguistics (ACL 2018) in the opposite side of the world: Melbourne. Interestingly, the combination of temporal proximity and geographical distance between these two conferences is becoming a tradition—last year it was ICML in Australia, and ACL in Canada.

This year, we presented one paper at ICML and another at ACL, and were invited to deliver a talk in the Second Workshop for Neural Machine Translation and Generation. This post shares some of my thoughts about both conferences.

ICML'18: Structured, Deep, Generative

Like NIPS, ICML is growing really fast, with 10 parallel tracks (it used to be 4 when I first attended ICML 10 years ago). Not surprisingly, a large fraction of the presented papers had to do with neural networks —their architecture, their training, their learned representations. A few

notes follow, with a focus on structured prediction, deep generative models, and representation learning.

View of central Stockholm from Södermalm, close to the ICML venue.

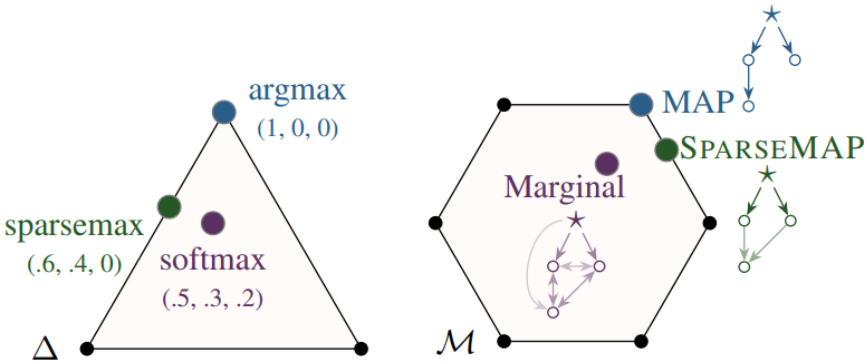
Structured Prediction

In the **Structured Prediction** track, Vlad Niculae presented our SparseMAP paper (a joint work with Mathieu Blondel and Claire Cardie)—a new technique for structured inference that outputs a **sparse** set of structures, as opposed to a single one (as in MAP inference) or a dense distribution over all structures (as in marginal inference).

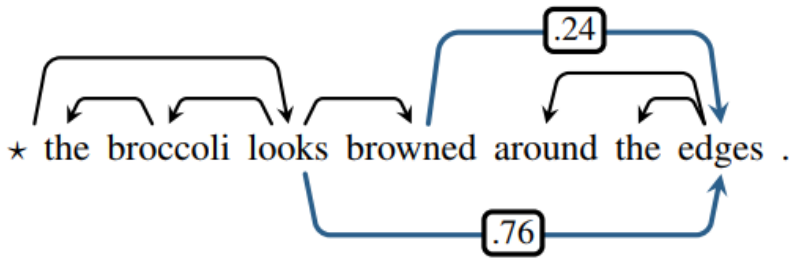
For example: the model may return only a handful plausible parse trees for a sentence, assigning zero probability to all the other, implausible, ones. SparseMAP is **differentiable** (we can plug it as a hidden layer in a neural network and run the usual gradient backpropagation) and **efficient** (thanks to an active set algorithm that evaluates SparseMAP by solving a sequence of MAP problems). We can regard it as the structured variant of sparsemax. Pytorch code is provided along with the paper.

Before Vlad, Nataly Brukhim gave a nice talk on "Modeling Cardinality in Structured Prediction", which plugs cardinality constraints for multi-label classification, handled through Dykstra's projection algorithm. Interestingly, I suspect this model can be regarded as a special case of our SparseMAP framework—the constraint corresponds to a "budget" factor, solvable in linear time and with closed-form gradient (likely

more efficient than their proposed approach of unrolling Dykstra's). Another related paper was presented the day after by Arthur Mensch: a way of using sparsemax in the update equations of dynamic programming algorithms, arriving at differentiable variants in-between sum-product and max-product.



SparseMAP as a structured version of sparsemax (figure extracted from here). Left: in the unstructured case (e.g. multi-class classification), sparsemax returns a sparse distribution over classes, represented as a point in the boundary of the simplex. Right: in the structured case, SparseMAP returns a sparse combination of structures (e.g. dependency parse trees), illustrated as a boundary point of the marginal polytope.



Example of two parse trees returned by SparseMAP for an ambiguous sentence. All other trees get zero probability (figure extracted from here).

Deep Generative Models

Deep generative models, most prominently variational auto-encoders (VAEs) and generative adversarial networks (GANs), are in great hype these days. Max Welling gave a beautiful invited talk ("Intelligence per Kilowatthour") motivating deep generative models through the lens of information theory, energy minimization, and the minimum description length principle, all rooted in the equation F (free energy) = E (energy)—H (entropy).

Several conference papers presented new insights in VAEs and GANs. VAEs are usually trained by maximizing the **evidence lower bound** (ELBO), a tractable variational approximation to the likelihood. It has

been observed, however, that this procedure often gives poor latent representations. As an alternative, Fixing a Broken ELBO proposes variational bounds of the **mutual information** between the input and latent variables, which are more general than ELBO. The result is a full rate-distortion curve that trades off compression and reconstruction accuracy, from which we can seek the Pareto optimal solutions.

Two other papers propose techniques to reduce the **amortization gap** of VAEs (this gap quantifies the suboptimality of the variational parameters found by inference networks, i.e., the encoder part of the VAE): Iterative Amortized Inference proposes an iterative strategy akin to learning-to-learn, which augments inference networks with an extra loop to update the variational parameters with stochastic gradients, while Semi-Amortized Variational Auto-Encoders proposes an hybrid approach between stochastic and amortized variational inference, leveraging differentiable optimization.

As for GANs, the central idea is, rather than approximating maximum likelihood training (equivalently KL divergence minimization), to replace it by a different objective: training the generator to create samples that **fool a supervised discriminator**. This boils down to a Jensen-Shannon divergence minimization problem (the original GAN paper) or the earth mover's distance (as in Wasserstein GANs).

An advantage of GANs over other generative models is that they tend to generate sharp outputs (useful when multi-modal distributions are desired); however, they also suffer from **mode collapse** (meaning that they tend to generate only from a few modes). Their training dynamics, which boil down to finding a Nash equilibrium in a minimax problem, is not yet fully understood. Lars Mescheder gave a nice presentation shedding some light on this, trying to answer the question Which Training Methods for GANs do actually Converge?, providing an analysis of when GANs converge by analyzing the spectrum of the Jacobian of the gradient field of the GAN objective.

Another challenge (very relevant to us NLP researchers) is to make GANs generate **discrete** data like text (most research on GANs has focused on continuous outputs like images). This is much more challenging, as the GAN training set-up, based on alternating gradient updates, requires the output of the generator to be differentiable, hence continuous. Previously proposed solutions include policy gradient methods like REINFORCE or the Gumbel-softmax reparametrization trick. Adversarially Regularized Autoencoders proposes an alternate solution, by combining a discrete auto-encoder with GAN-regularized

continuous latent representations, with interesting results in textual style transfer tasks. It feels, however, that there is still a lot to do in this space.

There was also this cool workshop on deep generative models that I unfortunately could not attend, as I was flying to ACL.

Sequence to Sequence

Sequence-to-sequence learning has been dominated by autoregressive models (e.g. in a recurrent decoder, each emitted output symbol is fed back as input to the LSTM in the next time step, creating a dependency on previous output symbols). **Non-autoregressive** sequence-to-sequence models are an active topic of research—if they worked, decoders could be parallelized and made much faster. Existing work accomplishes this, but typically with a drop in accuracy or the need for iterative refinement.

A Google AI paper, Fast Decoding in Sequence Models Using Discrete Latent Variables, proposes a **less autoregressive** approach, where a small number of latent variables are generated autoregressivelly (on top of a Transformer network), and then the target words are decoded in parallel, in a non-autoregressive manner, conditioned on the latent variables and the source words (there seems to be more recent work which combines this approach with distillation, with improved results). Apparently it is crucial that the latent variables are discrete, although I don't fully understand why.

Another machine translation paper, Analyzing Uncertainty in Neural Machine Translation, from FAIR, improves understanding of sequence-to-sequence models by assessing how uncertainty caused by noisy training data is captured by the model distribution and how it affects search, proposing tools to assess model calibration.

Representation Learning and Test of Time

Another interesting paper was Representation Tradeoffs for Hyperbolic Embeddings (and two other related papers: this and this), from which I got to learn about this recent framework of **embeddings in hyperbolic space**. The common theme in this (mathematically beautiful) line of research is that embeddings on hyperbolic manifolds (as opposed to the Euclidean space) are suitable for representing hierarchical relations among objects (for example, trees can be embedded in the Poincaré disk, the 2-dimensional hyperbolic space, with arbitrarily low distortion). As many objects of interest, e.g., social networks, WordNet-

like knowledge bases, entailment relations, etc. have a hierarchical structure, this geometry may be an appealing alternative to the usual flat Euclidean space. In this paper, they manage to represent the WordNet taxonomy with state-of-the-art precision with very compact two-dimensional representations.

Ronan Collobert and Jason Weston received the **Test-of-Time Award** for their influential ICML 2008 paper "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multi-Task Learning." I remember this paper very well (ICML 2008 was the first machine learning conference I attended), and the "cold" reception it got from most of the NLP audience, at a time where neural networks were everything but popular. I admit I was skeptical too—how could a single model, without any feature engineering, trained for weeks on the entire Wikipedia, achieve state-of-the-art scores in several NLP tasks?

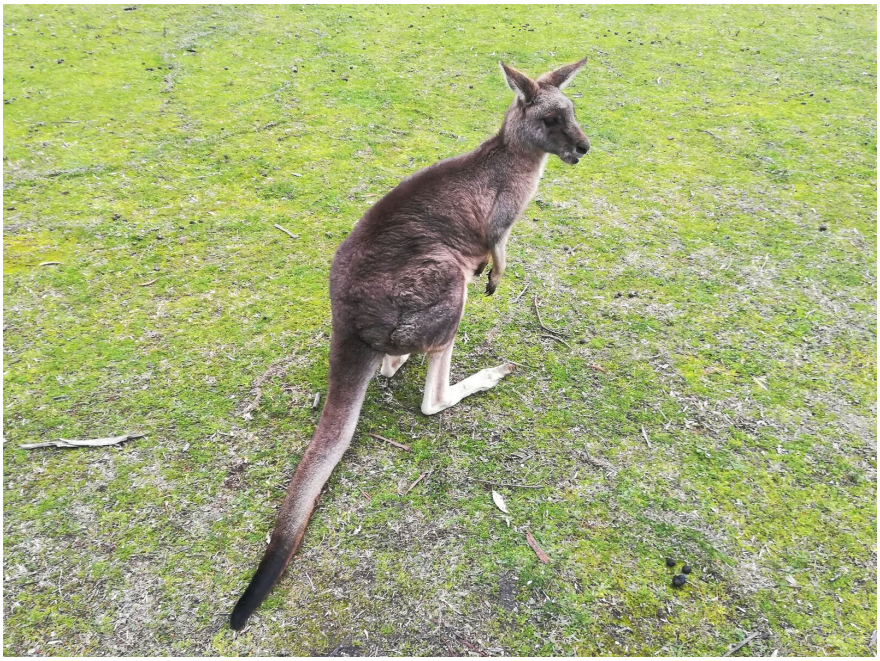
There were some problems in the original paper with the evaluation in the semantic role labeling task (fixed in a later JMLR paper) and a great deal of extrapolation from making progress in this shallow task to being close to solve "all of semantics" (This may be what annoyed the NLP community the most).

While the last problem persists in the community to this day, time has definitely proved that this was a really valuable contribution and the award very well deserved. Ten years later, we're all using continuous word representations trained on large datasets and training end-to-end models "from scratch." Except maybe for some industry niches, feature engineering is long gone and progressively replaced by **representation learning** and other forms of engineering: architecture search, hyperparameter tuning, transfer learning. Perhaps another take-home message (and I may lose some friends here) is that the NLP community should be more open to contributions from other fields, even when they seem ignorant about language—a closed community is condemned to overfitting their own techniques.

ACL'18: Fertility, Context-Aware Translation, Linguistic Structural Bias

After a few long flights and one extra day and a half on the calendar, I arrived in Melbourne for ACL. The conference was held in the Melbourne Convention and Exhibition Centre, a really nice venue (also one of the largest congress centers I've been to, rivaling with Sydney's International Convention Centre Sydney, which hosted ICML 2017).

The organization was spectacular. I hope more of these conferences come to Australia so that I have an excuse to go back.



Machine Translation and Generation

Chaitanya Malaviya presented our paper Sparse and Constrained Attention for Neural Machine Translation based on his internship work at Unbabel (joint work with Pedro Ferreira). The idea is to avoid some of the common mistake patterns in neural MT (under-translation and over-translation) by replacing the conventional softmax-based attention mechanisms by **constrained attention**: every time we generate a target word, we first upper bound how much attention each source word can receive (therefore avoiding repeatingly attending to the same words).

To this end, we keep track of how much cumulative attention each source word has received and use a **fertility-based approach** (akin to IBM models) to define the upper bound above. In addition, we encourage attention to be **sparse**, i.e. to give zero probability to unrelated words—this is accomplished via a novel **constrained sparsemax** transformation, which is cheap to compute and differentiable. We also propose new evaluation metrics to detect repetitions and dropped source words.

Chaitanya Malaviya presenting our ACL paper.

In Unsupervised Neural Machine Translation with Weight Sharing, a new architecture is proposed that, instead of a single encoder for both languages, uses separate encoders and a shared latent space, along with two GANs (a local and a global one) to enhance the cross-language translation.

Unsupervised NMT, though, appears to have some limitations, as stated in On the Limitations of Unsupervised Bilingual Dictionary Induction, presented by Sebastian Ruder: in particular, performance degrades considerably for morphologically rich languages (or simply when the source and target language are very dissimilar) and when embeddings are trained on different domains or using different algorithms.

This does not surprise me: I always found puzzling how bilingual embeddings can be induced from separate monolingual data by applying orthogonal (or more generally, linear) transformations. I understand the "local" structure of the embedding space is similar for both languages, but why should the "global" structure be similar? An orthogonal matrix has $O(D^2)$ degrees of freedom (D being the embedding size) but for practical problems the vocabulary size may well exceed D^2, so how can we have enough degrees of freedom in these rotations to superimpose the embeddings of related words? Maybe this works because the monolingual corpora are a bit parallel after all (which seems to be the case in Wikipedia), at least to the extent that the monolingual embeddings have similar structure to begin with. But this will break if their domain is different or if embeddings are learned with different algorithms.

A Google AI's paper, The Best of Both Worlds, carries out extensive experiments to disentangle several of the sources of Transformer Networks' improvement. They show that a "classic" recurrent NMT model on steroids, RNMT+ (which adopts some side innovations introduced in the Transformers paper, such as multi-head attention) is competitive to the non-recurrent attention-is-all-you-need architecture of the Transformer. Training may be tricky, though—Orhan Firat mentioned in the QA that things like layer normalization seemed crucial for stabilizing training both in Transformers and RNMT+.

A Stochastic Decoder for Neural Machine Translation addresses the fact that translation is a multi-modal, ambiguous process—there are typically many valid translations for a given sentence. They propose a VAE-like deep generative model of machine translation which allows sampling one of many-possible translations, by incorporating a chain of latent variables to account for lexical and syntactic variation in parallel corpora.

A really exciting research direction is in **context-aware NMT**—how to enable NMT models to work beyond the scope of a single sentence? This was the topic of Rico Sennrich's talk in the Second Workshop of Neural Machine Translation and Generation (more on Rico's talk below). There were several interesting talks in this topic, including Elena Voita's talk Context-Aware Neural Machine Translation Learns Anaphora Resolution, which allows the Transformer's encoder to condition on adjacent sentences (with a very interesting empirical analysis about where it learns to attend), and Sameen Maruf's talk on Document Context Neural Machine Translation with Memory Networks, which uses memory networks to store several sentences in a document and runs an iterative decoding algorithm based on block coordinate descent.

A few interesting papers concerning **text generation** include Style Transfer Through Back-Translation, from CMU, which proposes an adversarial latent variable model to rephrase text to contain specific stylistic properties (sentiment, gender and political slant), Hierarchical Neural Story Generation, from FAIR (awarded with an honourable mention), which learns a creative system to generate coherent and fluent stories given a short prompt (including a new dataset collected from Reddit), Generating Sentences by Editing Prototypes, from Stanford, which generates a sentence by sampling a prototype from the training set and refine it (a "semi-parametric" hybrid that mixes traditional left-to-right sentence generators and memory-based approaches), and Numeracy for Language Models: Evaluating and

Improving their Ability to Predict Numbers, from UCL, which looks at the ability of language models to accurately predict numerals—a current weakness of language models they propose to fix. Very interesting!

Parsing and Argmax Differentiation

Straight to the Tree from MILA proposes a new minimalistic constituent parser which predicts solely a vector of real-valued scalars ("syntactic distances"), for each split position in the input sentence, specifying a ranking order in which the split points will be selected—then a binary tree can be induced from this ranking, recursively, in a top-down fashion.

Stack-Pointer Networks for Dependency Parsing from CMU is an alternative approach to the traditional left-to-right transition-based parsers, using stack pointers for top-down parsing—at each point, the pointer picks the next left or right child to pick as modifier (like in a head automaton) and proceeds recursively. It's interesting to contrast this approach with transition-based parsers (which parse left to right) and easy-first parsers (which parse bottom up). Finally, the poster Constituency Parsing with a Self-Attentive Encoder from Berkeley gain a few extra points by replacing a BILSTM encoder with a self-attentive architecture (akin to Transformer networks), achieving some impressive 93.55 F1 in the PTB with a single model and no external data, and 95.13 F1 with the full model with ELMO embeddings.

Hao Peng from UW presented Backpropagating through Structured Argmax using a SPIGOT, a nice paper addressing the problem of how to backpropagate through the non-differentiable argmax operation, which was awarded an honorable mention. It proposes a variant of the straight-through estimator called SPIGOT—it involves computing the argmax in the forward pass, then computing an update in the backward that involves an Euclidean projection onto the marginal polytope.

Turns out this projection is **exactly** what SparseMAP computes too (see the ICML paragraph above), which makes these two approaches very related: while SparseMAP avoids argmax by computing this sparse projection on the forward step and then backpropagating with the exact gradient, SPIGOT computes the argmax in the forward step and then backpropagates a surrogate gradient. The building blocks are the same, though, so SPIGOT can also benefit from SparseMAP's active set algorithm to compute the Euclidean projection (making it usable for any problem where argmax is efficient). Overall, I prefer SparseMAP

though, as its training is likely more stable since it's based on the exact gradients (plus, it has a **sparse structured prediction** interpretation).

Another thought: it would be interesting to see how "marginal-SPIGOT" would perform (i.e. if we replace the Euclidean projection by a "KL projection," which often boils down to sum-product dynamic programming): "marginal-SPIGOT" would be to structured attention networks as SPIGOT is to SparseMAP.

Awards and Workshops

The last session was devoted to best paper awards and the ACL lifetime achievement award, given to Mark Steedman for his outstanding contributions to the areas of syntax and semantics, most noticeably the development of Combinatory Categorial Grammar.

Earlier, in Know What You Don't Know: Unanswerable Questions for SQuAD, winner of the best short paper award, a new version of the SQuAD dataset (a reference benchmark for **reading comprehension**) was introduced, combining the original questions with 50,000 unanswerable questions written adversarially by crowd-workers to look similar to answerable ones. This dataset seems much more challenging than the previous version, with a current gap of 20 F1 points between human and machine performance.

Queen Victoria Market in Melbourne.

In the workshop days, I attended Chris Dyer's talk on the Workshop on Relevance of Linguistic Structure in Neural NLP about "Combination

and Composition in Natural Language Learning." This was a beautiful talk inspired by Charles F. Hockett's "duality of patterning," which contrasts two levels of structure in language, namely how words are formed by **combining** distinctive and meaningless characters/phonemes, and how sentences are formed by **composing** significant and meaningful words. I'll focus on the composition process.

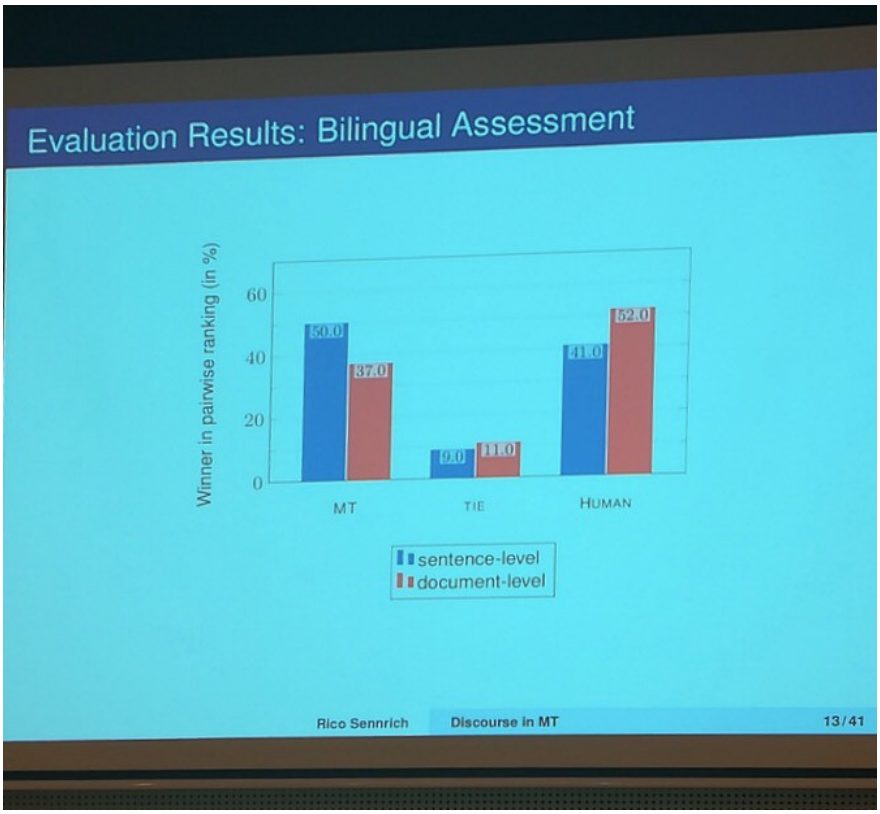
Why should we bother with modeling linguistic structure at all? Where do RNNs actually fail? Chris' talk showed two benefits of incorporating **structural inductive bias**: it makes our models more accurate (in particular, if we don't have a lot of data), and it helps us **understanding** the biases and limitations of non-structured models.

One way of spotting those limitations is to seek what linguistic information RNNs can encode (e.g. by trying to expose their representations into something human-readable like an FST—the topic of Yoav Goldberg's talk later in the same workshop). Chris' take is a bit different: to understand what RNNs can and can't do, compare the biases they learn with those of other structured models in terms of their "predictive preferences." A really compelling example are subject-verb agreement tasks, which require predicting the verb *is/are* in sentences like "He told me that the keys to the cabinet in the closet is/are on the table." As the number of "attractors" increases (i.e. nouns before the verb which are not the correct subject), models have an increasingly hard time figuring out what is the correct subject to pick the right verb form.

This experiment reveals the **sequential recency bias** of RNNs, which require too many hops (as many as surface words) to reach the subject from the verb, ending up learning an over-simplistic first-noun heuristics—picking *He* as the subject. By contrast, syntactically informed models such as RNN grammars, equipped with a stack data structure, develop a bias for **syntactic recency** which reduces the number of hops to get the subject right.

Finally, I attended the Second Workshop on Neural Machine Translation and Generation, where I gave an invited talk: Beyond Softmax: Sparsity, Constraints, Latent Structure—All End-to-End Differentiable! Here, I described various alternatives to softmax that can be used both in attention mechanisms and in output layers, including sparsemax, constrained softmax/sparsemax, and the most recent SparseMAP (see above), along with a unifying perspective of these transformations as induced by generalized entropies.

Rico Sennrich gave an interesting talk advocating Why the Time Is Ripe for Discourse in Machine Translation, with a nice historical perspective. A word of caution was given alluding to the recent Microsoft human parity achievement—in work in progress, a new **document-level evaluation** is being carried out which shows that, if the context is taken into account (as opposed to ranking translations in a sentence-by-sentence basis), then human parity is still far. Before Rico, Jacob Devlin described the several engineering tricks and architecture decisions to fit a NMT model in a cellphone. Related to this, there was a shared talk on Efficiency in NMT, where the Marian NMT system won in all settings (more information about Marian in this paper presented in the ACL demonstration session). The workshop closed with a talk from Yulia Tsetkov on how to get Flexible but Controllable Language Generation.



Document-level evaluation of MT: human parity not there yet (from Rico Sennrich's talk).

Take-Home Messages:

- Sparsemax/argmax differentiation are opening up new research fronts in structured prediction and NLP.
- Deep generative models are extremely promising, but still broken in many ways, and not yet able to handle discrete data

convincingly.

- Representation learning won the test of time in ML and NLP.
- Linguistic structure (fertility, syntactic phrases) are making its way as induction bias in end-to-end models.
- Machine translation is not solved, and the next big thing is how to take context into account, beyond sentence boundaries.

And that's all folks!

See you soon in Brussels for EMNLP.

Note: *I'd like to thank everyone at the Unbabel AI tribe and the DeepSPIN team members for giving feedback on this blog post and reading pre-prints of some of these papers in reading meetings. Other notes have been published on these conferences: check out Sebastian Ruder's ACL 2018 highlights and David Abel's detailed notes from ICML 2018, with a focus on reinforcement learning.*