Deep Auctions: Using Economics to Improve NMT Decoding

Stanford CS224N Custom Project

Logan Mondal Bhamidipaty
Department of Mathematics
Stanford University
loganmb@stanford.edu

Abhijit Devalapura
Department of Mathematics
Stanford University
abhydev5@stanford.edu

Abstract

We apply methods from auction theory to improve decoding for NMT. Current SOTA pipelines rely on greedy algorithms and sampling techniques that are rigid and produce homogeneous results. Although these methods are simple and fast and can sometimes be modified to produce more diverse outputs, they fail to reliably capture the richness of contextual embeddings on nuanced translation tasks. Our method provides an alternative strategy that produces higher-quality translations at the cost of computation speed.

1 Key Information to include

- Mentor: Hong Liu

2 Introduction

Problem Natural language is not inherently probability-maximizing (Holtzman et al., 2019). Native speakers avoid stating the obvious and try to convey meaningful, surprising information. Indeed, conversation would be very drab if the next word was always the most predictable one. Despite this, SOTA NMT resorts to greedy algorithms and simple probability-based sampling methods. Currently, beam decoding, top-p nucleus sampling and a handful of variants are the dominant methods. The general heuristic is to naively (or somewhat less naively in the case of top-p nucleus sampling and some beam variants) traverse a partial translation tree with some branching factor (a fixed hyperparameter k for naive beam or a threshold p for nucleus sampling). While these methods may work well in practice, there is certainly some theoretical room left for exploring alternative approaches. Our paper seeks to explore that realm through the paradigms of auction theory. We seek not to usurp the SOTA, but rather to demonstrate the viability and encourage the development of entirely novel techniques in NMT.

Overview At a high level, we model decoding as a competition between a set of bidders over a set of items. Bidders are token embeddings in candidate translations, and items are token embeddings in our source sentence. A network (GovNet) learns how much money to give each bidder (their "endowment") and how many items to sell at auction (the "item cap"). A different network that represents bidders (BidNet) takes in this information along with price and quantity demanded vectors (discussed below) in order to produce a value matrix that represents how much each bidder wants each item. We then feed this value information into a deterministic ascending auction (discussed below) that finds a tentative allocation of bidders to items (which we transform into a translation). If too many bidders want to buy the same item, the price of that item rises until a market clearing equilibrium is reached. At the end of each auction round, bidders update their values with new information they received at the end of the auction. *Through the auction, bidders compete with one another to win a place in the final translation.*

Motivation The relation between auction theory and machine translation is not obvious; however, if one probes with sufficient intention, a natural analogy emerges. For one, both machine translation and auctions represent complicated value dynamics. In NMT, certain words always go well together like "diagonalize" and "matrix"; "diagonalize" doesn't work well with any other subject. Other words are synonyms (e.g., one wouldn't say "She is my teacher instructor" even though "teacher" and "instructor" are similar; native speakers would probably only use either-or). Auctions also model this dynamic. Bidders could have value for a group of items because of inherent synergy (e.g., a bidder could have a high value for a pair of shoes but a low value for only a left sneaker) which is analogous to word complements. They could also view certain items as substitutes (e.g., if a bidder buys butter, they are less likely to value margarine and vice versa). This dynamic exists both over items and over other bidders (i.e., bidders care not only about *what* they win, but also about *who* else wins what).

3 Related Work

While we believe that the notion of applying auction theory to machine translation is entirely our own, there are of course many well-known alternative decoding strategies in NMT besides naive beam.

Diverse Beam Diverse beam is a variant of beam decoding that rewards diverse translation outputs (Vijayakumar et al., 2018). Although results proved promising at countering homogeneity in outputs, the method still relies heavily on tree-based decoding which ultimately limited the success of the technique.

Top-p Sampling Top-p sampling is another technique that improved naive beam decoding. By allowing the branching factor to vary as a function of probability, more diverse tokens could be sampled during decoding (Holtzman et al., 2019).

Auctions The most classic auction is called an ascending auction. As the name implies, prices start from zero and rise as long as multiple bidders find the price acceptable (i.e., their value for that item is at least as high as the price). Excess demand occurs when multiple bidders find a price and item acceptable (i.e., profitable), so the auctioneer increases prices until a market clearing equilibrium is reached. In theory, prices must rise continuously by some small ϵ, but in practice, sufficiently small, increments (we use $\epsilon = 0.01$).

One benefit of ascending auctions is that they overcome the so-called "exposure problem." Since prohibiting quantity reductions on items where prices don't increase helps in the case of substitutional preferences, it fails in the case of complements. That is why the CPA does not allow any bidder to reduce their quantity demanded for any item if the price has not increased on at least one item the bidder demanded in a previous round. This rule is implemented in our model.

Clock-Proxy Auctions (CPAs) CPA auctions are an impressive theoretical production. They hybridize ascending auctions with proxy auctions.[1].

There are various problems with classical auctions that we hypothesize could happen in translation decoding. This forms the basis of why the CPA was chosen to model after instead of a traditional auction design. One problem the CPA avoid is collusion. In regular auctions, bidders have a strong incentive to collude for obvious reasons. However, the clock phase of the CPA removes it as it limits bidder information to just excess demand for each item; the presence of aggregate information only is what solves the collusion difficulty.

The clock phase also gets around another canonical problem: the exposure problem. Since prohibiting quantity reductions on items where prices don't increase helps in the case of subsitutional preferences, it fails in the case of complements. That is why the CPA does not allow any bidder to reduce their quantity demanded for any item if the price has not increased on at least one item the bidder demanded in a previous round. This rule is implemented in our model.

The main problem the CPA avoids, however, is the "snake in the grass" problem. This title is given to the situation where bidders conceal their true interests from opponents by grossly understating their demands in early rounds, keeping the price low, and then creating a large price jump near the

[1]See: `https://en.wikipedia.org/wiki/Proxy_bid`

hypothesized end of the auction. The classical FCC rule does not work here, and neither does the monotonicity in quantity rule (both of which force bidders to have a downward-sloping demand curve) because when a variety of products are present, this can be too restrictive on the bidders preferences. So, the revealed preference activity rule, presented in (Ausubel and Milgrom, 2004) is used and implemented in our model. In short, what this rule allows is the following: Consider two times, s and t such that $t > s$. Allow p^s and p^t to be the price vectors at these times, and x^s and x^t be the associated demands of some bidder at those times, respectively. If $v(x)$ is the value of that bidder on package x, then define a sincere bidder as one who's values follow the two rules outlined below:

1. $v(x^s) - p^s \cdot x^s \geq v(x^t) - p^s \cdot x^t$
2. $v(x^t) - p^t \cdot x^t \geq v(x^s) - p^t \cdot x^s$.

The sum of these two inequalities shows the revealed preference activity rule (RP):

$$(p^t - p^s) \cdot (x^t - x^s) \leq 0.$$

The "snake in the grass," collusions, and the exposure problem are all eliminated in our model as each of the rules outlined above were implemented as stated. Our model takes inspiration from the CPA for its activity rule and anti-collusion mechanisms.

4 Approach

Our approach involves three phases: bidder and item generation, market preparation, and auctioning.

Bidder and Item Generation Recall that bidders are merely the embeddings of the tokens of the candidate hypothesis, and that items are simply embeddings from our source sentence. Our goal during generation is to intelligently create bidders and items from a source sentence. Generating items is simple: using an off-the-shelf model, we tokenize our source sentence and generate a matrix of contextual embeddings $I \in \mathbb{R}^{m \times d}$ where d is the embedding dimension of the model (we had $d = 512$).[2] The bidder tensor $B \in \mathbb{R}^{n \times d}$ is generated by creating n_1 candidate translations using a pre-trained model and then tokenizing the output using n_2 tokens and padding as needed. The result is that each source sentence becomes n_1 translations with n_2 tokens each, so the total number of bidders per example is $n = n_1 n_2$.

Market Preparation Before we begin the auction, we need to determine how much money every bidder has and how much of each item we will sell at the auction. GovNet decides both these quantities. It takes in input B and I, concatenating both tensors (since they share a common embedding dimension) and feeds the result to two separate stacks of fully-connected dense layers and a final projection layer that sizes the outputs: the endowment vector $e \in \mathbb{R}^n$ and the item cap vector $c \in \mathbb{R}^m$. We use ReLU as our activation since we want $e, c \succeq 0$.

Auctioning The auction is undoubtedly the most complicated portion of the entire model: it involves a wonderful amalgamation of boolean-relaxed convex optimization, auction theory and LSTMs.

BidNet takes in B and I as well as initial information about price $p \in \mathbb{R}^m$ and quantity demanded $q \in \mathbb{R}^n$ (quantity demanded refers to how many bidders want a certain item; we say that there is "excess demand" if quantity demanded exceeds the item cap i.e., $q \succ c$). We begin with every element of p being zero and every element of q being n (since everyone wants items that are free). BidNet uses $[B, I - (p \otimes q)\mathbb{1}^\top]$ as input into the LSTM. It projects the ouput of the LSTM into a tensor of values $V \in \mathbb{R}^{n \times m}$ where $V_{i,j}$ represents bidder i's value for item j. The values are then reported to the auction which determines an assignment χ by solving the boolean-relaxed convex optimization problem.[3]

$$\begin{aligned} \underset{\chi}{\text{maximize}} \quad & (V - p\mathbb{1}^\top) \otimes \chi \\ \text{subject to} \quad & \chi \in [0,1]^{n \times m} \text{ and } e \succeq \chi p \end{aligned}$$

[2] For clarity, we omit the batch size dimension in our report; however, the full model uses batches.

[3] See: `https://en.wikipedia.org/wiki/Linear_programming_relaxation`

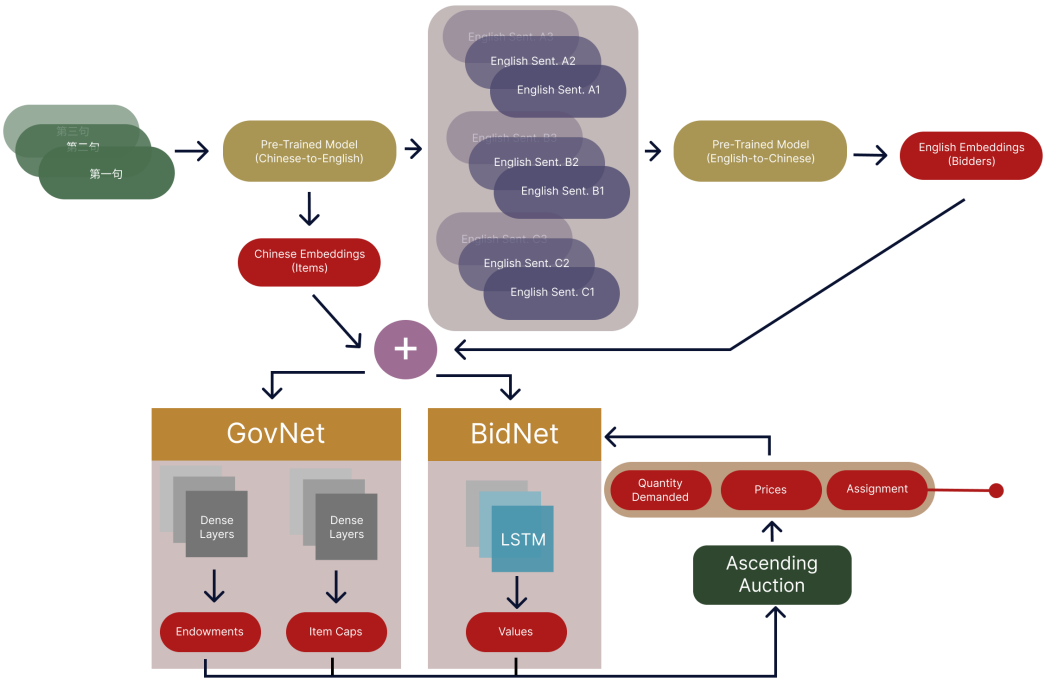

Figure 1: Diagram of Model Architecture

Once χ is determined, we update price and quantity demanded: $p = p + \epsilon \mathbb{1}_{q \succ c}$ and q is the row-wise sum of the elements in χ. These values are fed back into BidNet and the cycle repeats until $c \succeq q$; though, values are only allowed to be updated if the activity rule is satisfied (i.e., if the price of any good that the bidder has bid on increases, then the bidder is allowed to report lower values on the next round).

Note that we define a custom loss function for BidNet. First, we admit a hyperparameter α, representing the altruism of the bidders (how much they are concerned with self-interest, such as getting the allocation, versus how much they are concerned with overall translation quality). This is largely motivated by divergent behavior seen in real-world auctions. This hyperparameter is realized in the following way: we define a utility function as the following:

$$U(V, q, \chi) = \alpha \cdot \text{ translation quality } + (1 - \alpha) \cdot \mathbb{1}^\top (\chi \otimes (V - p\mathbb{1}^\top))\mathbb{1}$$

where translation quality is simply the categorical cross entropy of the predicted outputs which we can also extract through pre-trained models. This utility forms the negation of the loss function, and its important to realize that the less altruistic the bidders, the more they care about getting the allocation themselves, as shown by the direct product between the profit matrix and the allocations. GovNet is simply trained on translation quality. The reason these two models have different loss functions is because otherwise the market designer would collude with the bidders in the model (i.e., there would be no competition between bidders since they would optimize towards a single set of translations).

4.1 Data

We demonstrate the effectiveness of our approach on the WMT translation task from Chinese to English. Every year, WMT releases datasets (e.g. WMT17, WMT18, etc.) on several language pairs for a variety of tasks including translating biomedical, news, and other domain-specific documents. Since 2017, WMT began including data for Chinese and English translation tasks. We experimented with WMT19, focusing on general news-oriented tasks. The main WMT19 data set contains, as recommended by WMT in 2019, are the Europarl v10 parallel data set(wmt, 2023)(, 7th Frame-

work Programme), the UN corpus (Ziemsk, 2023) (wmt, 2023), and the news-commentary corpus (Tiedemann, 2012) (wmt, 2023). We tokenize our data using the MarianTokenizer.

4.2 Evaluation Method

We mainly used BLEU (as described in lecture and the class problem sets), CHaRacter-level F-score (chrF), sacreBLEU Post (2018), and COMET Rei et al. (2020) as recommended by WMT.wmt (2023) The chrF score computes commonalities between the output of a model and its reference translation by using character n-grams, which is especially useful for high-morphology languages. Although Chinese isn't considered a high-morphology language classically, we still include it as part of the standard evaluation suite recommended by WMT. SacreBLEU is very similar to the BLEU score; however, it provides more reproducible and comparable results. And, COMET produces values usually between $-\frac{3}{2}$ and 1 and is a neural evaluation framework using information from source and target reference translation sentences to improve machine translation accuracy.

4.3 Experimental Details

We used two subnetworks with three fully connected dense layers of forty units to produce e and c in GovNet. In BidNet, we used single LSTM with forty units. We use $\epsilon = 0.01$ and $p = 0.3$ with three sequence translations generated per input sentence. Those with more compute resources should experiment with high model parameters and lower values of p and ϵ. We trained for six hours on a personal GPU.

4.4 Results

Due to time constraints, we were only able to train our model on personal GPUs. As such, our results did not match SOTA performance (oce); however, we were able to produce meaningful training examples which are explored further in the analysis section.

Model	Place	BLEU	chrF	COMET (Normalized)	SacreBLEU
zh $\rightarrow$ en	$\#1$	33.5	0.6	-	-
zh $\rightarrow$ en	$\#2$	33.45	0.69	-	-
zh $\rightarrow$ en	-	25.1	0.501	0.63	62

Software limitations also severely handicapped the model. Since the auction phase uses convex optimization software, the model must be executed eagerly which slows down training. We expect that had we had more time to train, our model would have performed much better.

5 Analysis

Our model translated "1519年600名西班牙人在墨西哥登陆，去征服几百万人口的阿兹特克帝国，初次交锋他们损兵三分之二。" as "In 1519, six hundred Spaniards landed in Mexico to conquer the AzAztec Empire with a population of millions. They lost two thirdsin the first clash." It's interesting to note that the translation is quite accurate except for "AzAztec" and "thirdsin." Although these tokens are nonsensical, the overall translation structure is performs well. We believe that more training would dramatically improve our results. It succeeded at capturing nuanced phrase relations.

Interestingly, when asked to predict sentences with very little training data, the model was observed to commonly predict a high number unknown tokens. This matches our model architecture, because during bidder generation there are many similar bidders that we condense into a single representative. For example, if we were translating "我喜欢吃蛋糕," then the candidate translation before bidder segmentation would look like "I like cake", "I like pie", "I like cake", "I love cake". Since "I" always appears in the first position, we treat this token as identical to others and replace the other "I"'s with padding tokens to preserve order for the model. Because of this, there are many padded tokens in the bidder set. Because there are many padding tokens (the model treats padding and unknown tokens the same because it was shown to improve model performance), unknown tokens win early rounds of the auction and are common in early translations. In our analysis, we have observed that this phenomenon generally disappears after two to three hours of training on a local GPU.

6 Conclusion

Future work can proceed along several lines: (1) entropy-based sampling methods, (2) improved coalition dynamics, and (3) combinatorial auction models. Improving sampling methods improves bidder diversity. Unlike beam-based methods which must evaluate with a single (or diluted) metric, our model allows bidders to be generated via different streams. This opens the door for experimentation with more diverse bidder token generation potentially through entropy. Improved coalition dynamics and combinatorial decoding architectures may also yield promising results, but are outside the scope of our immediate work.

References

Wmt22: General mt task.

2023. Emnlp 2023 eighth conference on machine translation (wmt23).

Cramton Peter Ausubel, Lawrence M. and Paul Milgrom. 2004. The clock-proxy auction: A practical combinatorial auction design. pages 120–140.

European Commission (7th Framework Programme). 2012. European parliament proceedings parallel corpus 1996-2011.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search: Decoding diverse solutions from neural sequence models.

Junczys-Dowmunt M. Pouliquen B. Ziemsk, M. 2023. United nations parallel corpus v1.0.