Diversity Sampling in Machine Learning

Kalpesh Krishna (140070017) and Aryan Agal (16D170004)

IIT Bombay

Abstract

This project is a detailed study of [1], along with an implementation of [2] for the task of language generation using neural language models.

Diversity sampling is a general paradigm which attempts to discover different "modes" of a probability distribution. Diverse solutions are often very useful in situations where a user needs to choose a solution amongst M different options. Since the top M MAP solutions are usually very similar to each other and the MAP solution need not correspond to a suitable output, discovering a diverse set of solutions (M-best modes of the distribution) is a practical requirement. We study the paper [1], which poses this problem as a convex optimization problem, and proposes diversity sampling algorithms for three different problem domains - interactive segmentation, categorical segmentation and pose estimation.

Keeping up with recent trends in Deep Learning, the original diversity sampling technique proposed by [1] has been extended to neural beam search in [2, 3]. We investigate the benefits of [2] for language generation using neural language models.

Keywords: MRFs, Convex Optimization, Interactive Segmentation, Diverse Beam Search, Language Generation, Language Modeling

1. Code

The complete code for this project can be found at `https://github.com/martiansideofthemoon/diversity-sampling`.

2. Interactive Segmentation

Image segmentation labels an image, with each pixel being labeled as a foreground pixel or a background pixel. In interactive segmentation, the user annotates this image with scribbles specifying foreground and background regions of the image. This problem is often simplified by considering "super-pixels" (n x n grids of pixels) as the atomic unit of segmentation.

The probabilistic model is a Markov Random Field [4] (MRF) with each super-pixel being a vertex and an edge between adjacent super pixels in the grid. As a result, the largest cliques are edges. Each vertex x_i is a Bernoulli random variable, taking a foreground or background value. Energy functions are defined over each vertex and each edge.

$$\text{energy}(\mathbf{x}) = \sum_i \text{cost}(x_i) + \lambda \sum_{i,j} \text{cost}(x_i, x_j)$$

Here, $\text{cost}(x_i)$ refers to the cost of assigning a foreground or background value to pixel i, based on the scribbles provided by the user. Initially, a feature vector is learnt for each super-pixel by training a Transductive SVM [5] on the labeled super-pixels (scribbles). The model outputs a "score", which indicates the nature of the super-pixel. High positive scores indicate a tendency to be a foreground super-pixel whereas low negative scores indicate a tendency to be a background super-pixel. [6] contains the exact formulation of this baseline.

$\text{cost}(x_i, x_j)$ is a smoothness term, which uses the contrast sensitive Potts model formulation.

$$\text{cost}(x_i, x_j) = I(x_i \neq x_j) \cdot c_1 \cdot e^{-c_2, d_{ij}}$$

Here c_1 and c_2 are positive constants, and d_{ij} indicates the similarity between the superpixel feature vectors. Similar adjacent superpixels with different labels are penalized more heavily in this formulation.

The contrast-sensitive Potts model is a submodular energy function for a binary labeling problem so we compute the MAP solution using the graph-cut solution in [7].

3. Diversity Sampling

The diversity sampling algorithm is applied to the MAP problem on MRFs which attempts to find,

$$\mathbf{x}^* = \min_{\mathbf{x}} \sum_i \text{cost}(x_i) + \sum_{i,j} \text{cost}(x_i, x_j)$$

This formulation is identical to the interactive segmentation problem, since λ can be subsumed into the pairwise potentials. This problem is often converted into a MAP integer program. Let $\boldsymbol{\theta}_i$ denote the vector of all possible energy values of x_i, and $\boldsymbol{\mu}_i$ be a one-hot vector. Hence, $\text{cost}(x_i) = \boldsymbol{\theta}_i \cdot \boldsymbol{\mu}_i$. Similarly, we can define an integer program for edge potentials. This results in an NP-hard integer program,

$$\mathbf{x}^* = \min_{\boldsymbol{\mu}} \sum_A \boldsymbol{\theta}_A \cdot \boldsymbol{\mu}_A$$

Subject to linear constraints ensuring $\boldsymbol{\mu_A}$ is consistent across edges and a one-hot indicator vector taking on values 0 or 1.

Once we compute a MAP solution, a diverse 2^{nd} mode can be computed using a dissimilarity function Δ. Let $\boldsymbol{\mu}^k$ denote a full configuration set of all $\boldsymbol{\mu}_A$ values. Let $\boldsymbol{\mu}^1$ be the MAP solution. We add a constraint to the above MAP problem, $\Delta(\boldsymbol{\mu}^1, \boldsymbol{\mu}) \geq k$ (the solution is sufficiently "far away" from the MAP). This approach can be extended iteratively, to find a m^{th} diverse solution, atleast k_m units away from each of the previous $m-1$ solutions.

3.1. Lagrangian Relaxation of MAP Integer Program

The diversity formulation is at least as hard as the original MAP program. A continuous relaxation of the problem is studied, with a relaxed constraint $\boldsymbol{\mu}_A \succeq 0$. The diversity constraints are dualized (with multipliers $\boldsymbol{\lambda}$) to form,

$$f(\boldsymbol{\lambda}) = \min_{\boldsymbol{\mu}} \sum_A \boldsymbol{\theta}_A \cdot \boldsymbol{\mu}_A - \sum_1^{m-1} \lambda_i(\Delta(\boldsymbol{\mu}, \boldsymbol{\mu}^i) - k_i)$$

For all $\lambda \geq 0$, this is a lower bound of the primal solution [8]. Hence, the best solution would be obtained by finding $\max_{\boldsymbol{\lambda} \succeq 0} f(\boldsymbol{\lambda})$. Since $f(\boldsymbol{\lambda}) =$

$\min_{\boldsymbol{\mu}} \mathbf{a}_{\boldsymbol{\mu}} \cdot \boldsymbol{\lambda} + b_{\boldsymbol{\mu}}$, it is a non-smooth function in general. It is also concave in $\boldsymbol{\mu}$. We can see this as follows,

$$\begin{aligned} f(k\boldsymbol{\lambda}_1 + (1-k)\boldsymbol{\lambda}_2) &= \min_{\boldsymbol{\mu}} \{k(\mathbf{a}_{\boldsymbol{\mu}} \cdot \boldsymbol{\lambda}_{\mathbf{1}} + b_{\boldsymbol{\mu}}) + (1-k)(\mathbf{a}_{\boldsymbol{\mu}} \cdot \boldsymbol{\lambda}_{\mathbf{2}} + b_{\boldsymbol{\mu}})\} \\ &\geq k \min_{\boldsymbol{\mu}} (\mathbf{a}_{\boldsymbol{\mu}} \cdot \boldsymbol{\lambda}_{\mathbf{1}} + b_{\boldsymbol{\mu}}) + (1-k) \min_{\boldsymbol{\mu}} (\mathbf{a}_{\boldsymbol{\mu}} \cdot \boldsymbol{\lambda}_{\mathbf{2}} + b_{\boldsymbol{\mu}}) \\ &= kf(\boldsymbol{\lambda}_1) + (1-k)f(\boldsymbol{\lambda}_2) \end{aligned}$$

Due to the non-smooth concave nature, a supergradient ascent algorithm is used to maximize the dual (which converges to the maxima whenever suitable step-sizes are chosen, $\lim_{t\to\infty} \alpha_t = 0$ and $\sum_t \alpha_t = \infty$). Since $f(\boldsymbol{\lambda})$ is a point-wise minima of several functions, its super-gradient is $\mathbf{a}_{\boldsymbol{\mu}^*}$, where $\boldsymbol{\mu}^* = \arg\min_{\boldsymbol{\mu}} \mathbf{a}_{\boldsymbol{\mu}} \cdot \boldsymbol{\lambda} + b_{\boldsymbol{\mu}}$. This can be shown as follows for any $\boldsymbol{\lambda}_2$.

$$\begin{aligned} \mathbf{a}_{\boldsymbol{\mu}^*} \cdot \boldsymbol{\lambda}_2 + b_{\boldsymbol{\mu}^*} &\geq \min_{\boldsymbol{\mu}} \{\mathbf{a}_{\boldsymbol{\mu}} \cdot \boldsymbol{\lambda}_2 + b_{\boldsymbol{\mu}}\} \\ \mathbf{a}_{\boldsymbol{\mu}^*} \cdot (\boldsymbol{\lambda}_2 - \boldsymbol{\lambda}) &\geq \min_{\boldsymbol{\mu}} \{\mathbf{a}_{\boldsymbol{\mu}} \cdot \boldsymbol{\lambda}_2 + b_{\boldsymbol{\mu}}\} - \mathbf{a}_{\boldsymbol{\mu}^*} \cdot \boldsymbol{\lambda} - b_{\boldsymbol{\mu}^*} \\ \mathbf{a}_{\boldsymbol{\mu}^*} \cdot (\boldsymbol{\lambda}_2 - \boldsymbol{\lambda}) &\geq f(\boldsymbol{\lambda}_2) - f(\boldsymbol{\lambda}) \end{aligned}$$

Which is the condition for $\mathbf{a}_{\boldsymbol{\mu}^*}$ to be a super-gradient. Hence we obtain the super-gradient of $f(\boldsymbol{\lambda})$ as,

$$\nabla_i f(\boldsymbol{\lambda}) = -(\Delta(\boldsymbol{\mu}^*, \boldsymbol{\mu}^i) - k_i)$$

Here $\boldsymbol{\mu}^*$ is the optimal solution for a fixed value of $\boldsymbol{\lambda}$. This solution is intuitive since if $\Delta(\boldsymbol{\mu}^*, \boldsymbol{\mu}^i) \leq k_i$, the gradient will be positive, thus increasing the penalty due to violation of this constraint in the next iteration (and encouraging the model to satisfy the constraint).

3.2. Computing the Super-gradient

We use a strategically chosen Δ function to reuse the original MAP formulation. More specifically, we use the Hamming distance (offset by a constant), where $\Delta(\boldsymbol{\mu}^1, \boldsymbol{\mu}^2) = -\sum_{i\in\text{vertices}} \boldsymbol{\mu}_i^1 \cdot \boldsymbol{\mu}_i^2$. This formulation makes it easy to include this term in the unary potentials, giving us a final formulation,

$$\begin{aligned} f(\boldsymbol{\lambda}) &= \min_{\boldsymbol{\mu}} \sum_{A} \boldsymbol{\theta}_A \cdot \boldsymbol{\mu}_A - \sum_{i=1}^{m-1} \lambda_i (-\sum_{j \in V} \boldsymbol{\mu}_j \cdot \boldsymbol{\mu}_j^i - k_i) \\ &= \min_{\boldsymbol{\mu}} \sum_{j \in V} (\boldsymbol{\theta}_j + \sum_{i=1}^{m-1} \lambda_i \boldsymbol{\mu}_j^i) \cdot \boldsymbol{\mu}_j + \sum_{j \in E} \boldsymbol{\theta}_j \cdot \boldsymbol{\mu}_j - \sum_{i=1}^{m-1} \lambda_i k_i \\ &= \min_{\boldsymbol{\mu}} \sum_{j \in V} \boldsymbol{\theta}'_j \cdot \boldsymbol{\mu}_j + \sum_{j \in E} \boldsymbol{\theta}_j \cdot \boldsymbol{\mu}_j - \sum_{i=1}^{m-1} \lambda_i k_i \end{aligned}$$

The last term is a constant which won't affect the argmax solution of $\boldsymbol{\mu}$ for a constant $\boldsymbol{\lambda}$. Hence we can utilize the same MAP machinery to compute the desired super-gradients, with modified energy functions.

3.3. Tightness of the Dual

The dot product formulation leaves a duality gap in the formulation [9], but experimental evidence shows that it works well in practice. The M-best MAP solution does not leave a duality gap. This is proved in [1].

4. Interactive Segmentation Results

We utilize a MATLAB implementation[1] of [1] for interactive segmentation and obtain results consistent with the theoretical analysis presented in the previous sections. We try to present results not shown in [1] in Figure 1,2,3 and 4.

[1] https://github.com/batra-mlp-lab/divmbest

Figure 1: The third, fourth and fifth diverse solutions fill up errors in the segmentation of the man's head

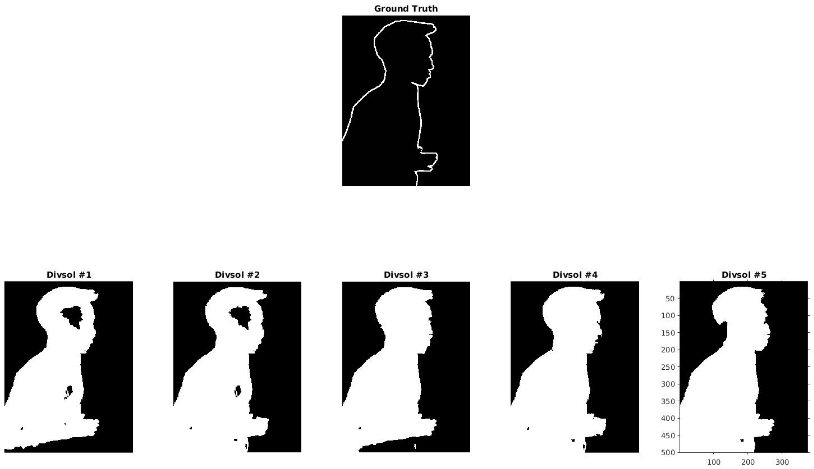


Figure 2: The third diverse solution is able to model the thin mast of the boat

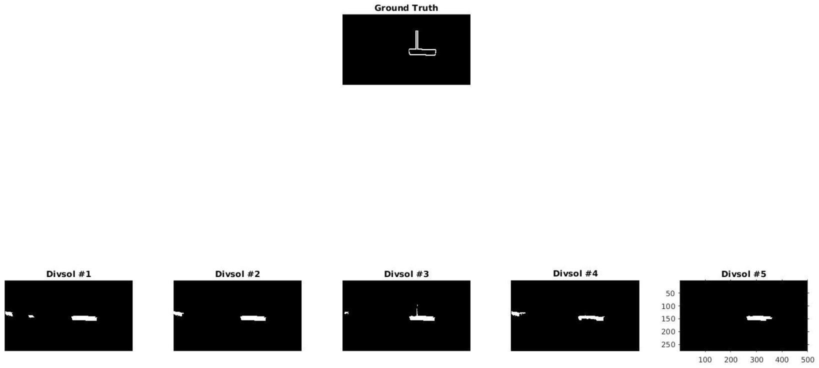


Figure 3: The fifth diverse solution fill up errors outside object of focus

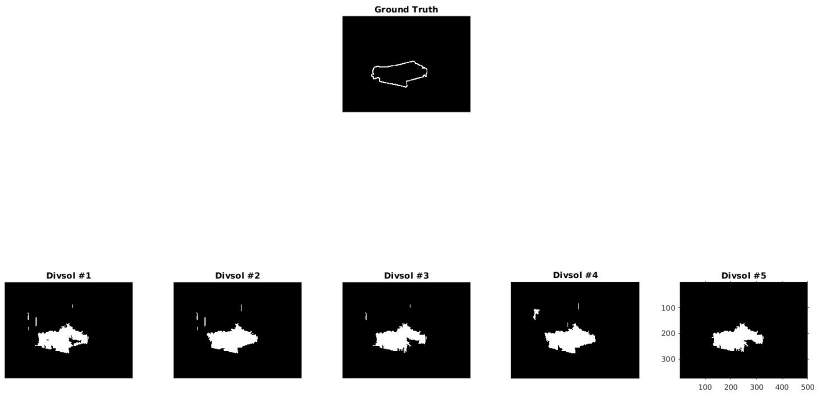
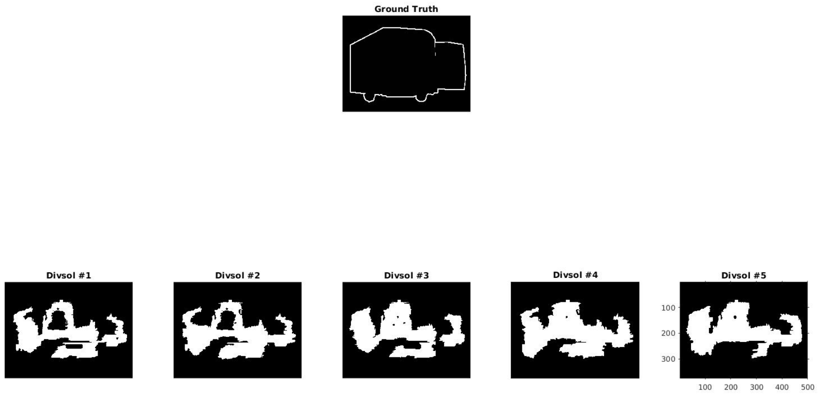


Figure 4: The fourth and fifth diverse solutions attempt to fill up the truck area of focus



5. Diversity Sampling for NLP

Shortly after [1], there has been some work in diversity sampling for natural language processing tasks. Most notably, [10] suggests an n-gram penalty function used for diverse phrase-based non-neural machine translations. [2] explores diverse neural beam search for the tasks of visual question generation and image captioning. [3] explore diverse neural beam search for dialogue response generation, abstractive summarization, and machine translation.

In this work we experiment with the simpler, more fundamental task of language generation using neural language models in a generative sense.

6. Neural Language Models

Language models are language-specific models used to model the probability of a sentence, $p(w_1, w_2...w_n)$. A specific `<EOS>` is introduced to mark the end of sentences to allow the model to predict probabilities of sentences varying in length [11]. Language models can be treated as "next-word predictors" by splitting the joint distribution into conditional probabilities,

$$p(w_1, w_2...w_n) = p(w_1)p(w_2|w_1)p(w_3|w_1w_2)...p(w_n|w_1, w_2...w_{n-1})$$

Neural language models are often evaluated using perplexity, which measures the average uncertainty for a next-word prediction. A perplexity of 100 indicates that on an average the model is unsure between 100 equally likely next-words. Perplexity is computed using,

$$\text{ppl} = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2 p(w_i|w_1,w_2...w_{i-1})}$$

As our baseline language model, we use a 650-unit 2-layer LSTM based on [12]. This achieves a decent perplexity of 78 on Penn Treebank [13] (While the current state-of-the-art hovers close to 48, we choose this model due to its simplicity and strong performance. This was the state-of-the-art in 2014).

7. Diverse Beam Search in Language Models

We implement the beam search algorithm mentioned in [2]. We assume a single group of size B (beam width). We adopt a beam size B of 10 and record beams up to the first `<EOS>` occurrence. We perform the following diversity sampling algorithm (see equations in [2]),

```
for $t = 1...T$ do
    // Decode first beam element without diversity;
    for $b = 2...B$ do
        // Augment log-probabilities with diversity penalty;
        // Perform one step of decoding for $b^{th}$ element;
    end
end
```

Algorithm 1: Diverse Beam Search

We present an analysis on two different sentence-level difference schemes. Our first scheme is a simple difference between the lengths of the decoded sentences. We want to encourage our model to produce sentences greatly varying in length. We define our difference as,

$$\Delta(y_1, y_2) = |\text{len}(y_1) - \text{len}(y_2)|$$

We initialize beams with a prior (to feed the LSTM hidden states with meaningful semantic information) and sample the `argmax` output at each time-step of the beam search decoding. Our results are presented in Table 1. We notice that our sentences are largely preserving their prefixes, but the lengths greatly vary due to the diversity of the length.

Table 1: Length difference function results. λ denotes the weight assigned to the diversity score relative to the log probability of the generated sentence so far. `EOS` denotes the end of the sentence. The different sentences correspond to unique sentences in the beam.

λ	Beam #	Prior	Decoded
1	1	he is a lawyer	for the company `EOS`
	2	he is a lawyer	`EOS`
1	1	how can i do	it `EOS`
	2	how can i do	`EOS`
5	1	he is a lawyer	for the company `EOS`
	2	he is a lawyer	lawyer `EOS`
	3	he is a lawyer	lawyer for the company 's board of directors and a director of the national association of securities dealers and the exchange 's management committee and the board of trade and industry 's board of trade `EOS`
	4	he is a lawyer	lawyer for the company 's board of directors and a director of the national association of securities dealers and the exchange 's management committee `EOS`
	5	he is a lawyer	lawyer for the company 's board of directors and a director of the national association of securities dealers `EOS`
5	1	how can i do	it `EOS`
	2	how can i do	`EOS`
	3	how can i do	it again says mr. verwoerd `EOS`
	4	how can i do	it again says mr. verwoerd who is a member of the journal 's new york bureau `EOS`

As a second difference function, we investigate the Hamming distance. The Hamming distance on a word level is modeled as the number of differing words at the same location. To avoid sentence length diversity, we define our difference as,

$$\Delta(y_1, y_2) = \sum_{i=1}^{\min\{l_1,l_2\}} [[y_{1i} \neq y_{2i}]]$$

Here $[[a \neq b]] = 1$ when the condition is true, otherwise 0. We present our results in Table 2. As evident from the results, for $\lambda = 1$ we succeed in getting some diversity, but we have a lot of repetition across beams (for $\lambda = 0$ we would get all 10 identical beams. However, for $\lambda = 5$, we get very different sentences for all 10 beams, indicating successful Hamming distance diversity. We observe no significant differences in length as expected.

Table 2: Hamming difference function results. λ denotes the weight assigned to the diversity score relative to the log probability of the generated sentence so far. EOS denotes the end of the sentence. Here the "Beams" column represent the number beams having that particular decoded transcript.

λ	# Beams	Prior	Decoded
1	5	why the	company has been able to spend more than $ N million in cash and $ N million in assets EOS
	4	why the	company has been able to sell the company 's N N stake in the company EOS
	1	why the	government is n't likely to be able to get the money to the public EOS
5	1	why the	company has been able to spend more than $ N million in cash and $ N million in assets EOS
	1	why the	government is n't likely EOS
	1	why the	u.s. government is n't allowed to take the action EOS
	1	why the	new york stock exchange is n't likely to be the only way to sell the stock EOS
	1	why the	two companies are in the best position of the company EOS
	1	why the	N N of the N N of them are in the N model year EOS
	1	why the	federal reserve will be able to make a new bid for the company EOS
	1	why the	market was in a severe crunch EOS
	1	why the	bank 's assets are n't the only way to be a good investment EOS
	1	why the	state department has been investigating whether the u.s. has been able to spend the money in the u.s. EOS

8. Conclusion

We present a detailed analysis on diversity sampling techniques used in machine learning and present diverse results for interactive segmentation and language generation using neural language models. Future work in this space includes more innovative difference functions for NLP tasks, (such as the n-gram difference introduced by [10], or BLEU scores) and more robust algorithms to enforce diversity.

9. References

[1] D. Batra, P. Yadollahpour, A. Guzman-Rivera, G. Shakhnarovich, Diverse m-best solutions in markov random fields, in: European Conference on Computer Vision, Springer, pp. 1–16.

[2] A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. Crandall, D. Batra, Diverse beam search: Decoding diverse solutions from neural sequence models, arXiv preprint arXiv:1610.02424 (2016).

[3] J. Li, W. Monroe, D. Jurafsky, A simple, fast diverse decoding algorithm for neural generation, arXiv preprint arXiv:1611.08562 (2016).

[4] R. Kindermann, J. L. Snell, Markov random fields and their applications, volume 1, American Mathematical Society, 1980.

[5] V. Sindhwani, S. S. Keerthi, Large scale semi-supervised linear svms, in: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 477–484.

[6] P. Yadollahpour, D. Batra, G. Shakhnarovich, Diverse m-best solutions in mrfs, in: Workshop on Discrete Optimization in Machine Learning, NIPS.

[7] P. Kohli, P. H. Torr, Efficiently solving dynamic markov random fields using graph cuts, in: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, volume 2, IEEE, pp. 922–929.

[8] S. Boyd, L. Vandenberghe, Convex optimization, Cambridge university press, 2004.

[9] A. M. Geoffrion, Lagrangean relaxation for integer programming, in: Approaches to integer programming, Springer, 1974, pp. 82–114.

[10] K. Gimpel, D. Batra, C. Dyer, G. Shakhnarovich, A systematic exploration of diversity in machine translation, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1100–1111.

[11] S. F. Chen, J. Goodman, An empirical study of smoothing techniques for language modeling, Computer Speech & Language 13 (1999) 359–394.

[12] W. Zaremba, I. Sutskever, O. Vinyals, Recurrent neural network regularization, arXiv preprint arXiv:1409.2329 (2014).

[13] M. P. Marcus, M. A. Marcinkiewicz, B. Santorini, Building a large annotated corpus of english: The penn treebank, Computational linguistics 19 (1993) 313–330.