Appendix for Putting Back the Stops: Integrating Syntax with Neural Topic Models

Author Name
Affiliation
email@example.com

A Proofs and Training

In this section, we present the proof for Theorem 3.1, ultimately leading to the objective of the C-VAE model.

For clarity, the plate notation for C-VAE is given in Figure 1. In the figure, x_i represents part of the data from feature class i (e.g. x_1 might be the semantic words, and x_2 the syntactic words). Each x_i is a BoW vector and $\sum_i x_i = x$ is one document. z_i are the corresponding latent vectors. C represents the context network which decides how each document is divided into n parts $x_i, \ldots, x_n$. Assuming the data features are divided into n parts, where each part is considered independent, the log-likelihood of the data can be given as:

$$\log\ p(x) = \log \prod_{i=1}^{n} p(x_i)$$

Using the Bayes' rule to rewrite the log-likelihood in terms of the latent variables:

$$\log\ p(x) = \log \int \prod_{i=1}^{n} p(x_i|z_i)p(z_i)\, dz_1 \ldots dz_n$$

Introducing the approximate posterior distributions for $z_1, z_2, \ldots, z_n$:

$$\log\ p(x) = \log \int \prod_{i=1}^{n} q(z_i|x)p(x_i|z_i)p(z_i) \cdot \prod_{i=1}^{n} \frac{1}{q(z_i|x)}\, dz_1 \ldots dz_n$$

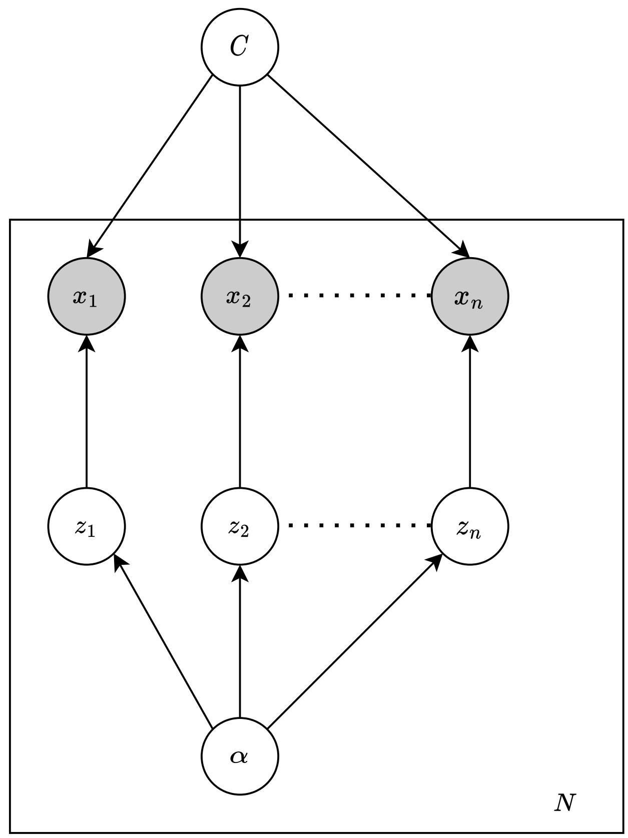

Figure 1: C-VAE in plate notation

Using the property of the log function to split the integral:

$$\log\ p(x) = \sum_{i=1}^{n} \mathbb{E}_{q(z_i|x)}[\log\ p(x_i|z_i)] + \sum_{i=1}^{n} \log \int q(z_i|x)p(z_i)\, dz_1 \ldots dz_n$$

Substituting the KL-divergence for the last term:

$$ELBO = \sum_{i=1}^{n} \mathbb{E}_{q(z_i|x)}[\log\ p(x_i|z_i)] - \sum_{i=1}^{n} D_{\text{KL}}(q(z_i|x)\|p(z_i))$$

Algorithm 1 Training SyConNTM

Input: documents (D), context type (C_{type}), and size (c), decision type dt, and decision hparam t/M
Initialize hyper-parameters and model parameters.
for batch $\mathcal{B}$ in D **do**
  Get x_{BoW} and x_{seq} from $\mathcal{B}$
  $s = \text{context\_net}(x_{\text{seq}}, C_{\text{type}}, c, E)$
  Update parameters (Equation 2)
  $x^{\text{syn}}_{\text{BoW}}, x^{\text{sem}}_{\text{BoW}} = \text{dec\_module}(s, \text{dt}, t/M, x_{\text{seq}}, x_{\text{BoW}})$
  $x^{\text{syn}'}_{\text{BoW}}, x^{\text{sem}'}_{\text{BoW}}, z_{\text{syn}}, z_{\text{sem}} = \text{SC-NTM}(x_{\text{BoW}})$
  Update SC-NTM parameters (Equation 4)
end for

Re-writing $ELBO$ in terms of Equation 1:

$$\mathcal{L}_{\text{c-vae}}(\theta, \phi; x) = \sum_{i=1}^{n} \mathcal{L}(\theta, \phi_i; x)$$

A.1 Training:

The algorithm for training SyConNTM is shown in Algorithm 1. The input to the algorithm includes a set of documents (D), a context type (C_{type}) and context size c for the context network, and a decision type dt for the decision module. For each batch $\mathcal{B}$ in D, a BoW representation, x_{BoW}, and a sequential representation x_{seq}, is constructed. These representations are then passed through the $context\_net$ (Algorithm 3, Appendix B) which gives probabilities s for each word in x_{seq}. The parameters of the context network are updated based on Equation 2. The prediction probabilities s are then passed through the dec_module (Algorithm 4, Appendix C) which generates two bag-of-word vectors, $x^{\text{syn}}_{\text{BoW}}$ and $x^{\text{sem}}_{\text{BoW}}$ with syntax and semantic words respectively. Finally, x_{BoW} is passed through the SC-NTM model, which updates its parameters based on Equation 4. We also investigated pretraining the context network which resulted in a faster convergence of SyConNTM at the cost of the pretraining overhead, but did not improve results.

A.2 Generative process of the network:

We describe the generative process of a document in Algorithm 2 using a combination of semantic and syntactic topics. Firstly, for a given document, we sample the semantic topic distribution, denoted as $\theta_d^{semantic}$, from a Dirichlet distribution with parameter α. This distribution captures the distribution of semantic topics within the document. Similarly, we sample the syntactic topic distribution, $\theta_d^{syntactic}$, using the same procedure. Next, for each word in the document, we draw a class, c_i, from the context_net, represented by $\pi_{context}$. If the class is determined to be of the semantic type ($c_i = 1$), we draw a topic, z_i, from the semantic topic distribution. Conversely, if the class is syntactic, we draw a topic, z_i, from the syntactic topic distribution. Finally, we draw the word, w_i, from the word distribution ϕ_{z_i}, which is parameterized by the VAE decoder. This generative process allows us to generate documents and capture both semantic and syntactic characteristics.

Algorithm 2 Generative process of the network

Sample semantic topic distribution for document d, $\theta_d^{semantic} \sim Dirichlet(\alpha)$
Sample syntactic topic distribution for document d, $\theta_d^{syntactic} \sim Dirichlet(\alpha)$
for each word w_i in D **do**
  Draw class $c_i \sim \pi_{context}$ (here π comes from the context_net)
  if $c_i = 1$ (semantic class) **then**
    Draw topic $z_i \sim \theta_d^{semantic}$
  else
    Draw topic $z_i \sim \theta_d^{syntactic}$
  end if
  Draw $w_i \sim \phi_{z_i}$ (word distributions in ϕ are parameterized by the VAE decoder)
end for

B Context Network

The context network is similar to the feed forward n-gram model by [Chelba *et al.*, 2017] and takes as input a concatenation of embeddings of c context words, i.e. $X = \text{concat}(E(w_{i-(c-1)}), \ldots E(w_{i-1}))$.

Our context network is defined as folllows:

$$X_1 = \text{dropout}(X, p_{\text{keep}}) \quad (1)$$

$$X_2 = \tanh(L^1 \cdot X_1 + L^1_{\text{bias}}) \quad (2)$$

$$X_3 = \text{BatchNorm}(\text{dropout}(X_2, p_{\text{keep}})) \quad (3)$$

$$\text{out} = \text{softmax}(L^2 \cdot X_3 + L^2_{\text{bias}}) \quad (4)$$

The above equations represent the context network in a series of operations that are performed on the input data, denoted by the variable X. The first operation in Equation 1 applies the dropout regularization technique [Srivastava *et al.*, 2014] to the input data, where p_{keep} is the probability of retaining a given input element during the dropout operation. The second operation in Equation 2 applies the hyperbolic tangent activation function to the dot product of the first layer weights, L^1, and the dropout-regularized input data, X_1, plus the first layer bias term, L^1_{bias}. The third operation in Equation 3 applies batch normalization [Ioffe and Szegedy, 2015] and dropout regularization to the output of the previous operation, X_2. Finally, the fourth operation in Equation 4 applies the softmax activation function to the dot product of the second layer weights, L^2, and the batch-normalized, dropout-regularized input, X_3, plus the second layer bias term, L^2_{bias}. The output of the final operation, out, represents the model's prediction. The hyperparameters for the model are: p_{keep} and the dimensionality of the weight vectors L^1 and L^2.

The context network predicts a target word w_i given a context size c. We experiment with two types of context: symmetric $(w_{i-(c-1)}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+(c+1)})$ and asymmetric $(w_{i-(c-1)}, \ldots, w_{i-1})$ for a target word w_i.

We define the Algorithm 3 for generating target probability r and predicted probability s from an input sequence x_{seq} of length n, a context type C_{type}, a context size C_{size}, and an embedding look-up table E. s is initialized with zero to store the prediction probabilities. The algorithm iterates over each

Algorithm 3 context_net

Input: x_{seq} of length n, context type C_{type}, context size c, embedding look up table E
Return: prediction probability s
Initialize $s \in \{0\}^{n \times V}$ as an empty array.
for target words $w_i, \ldots, w_n$ **do**
  if C_{type} == 'symmetric' **then**
    $cv \leftarrow \text{concat}(E(w_{i-(c-1)}), \ldots, E(w_{i-1}), E(w_{i+1}), \ldots, E(w_{i+(c+1)}))$
  else if C_{type} == 'asymmetric' **then**
    $cv \leftarrow \text{concat}(E(w_{i-(c-1)}), \ldots, E(w_{i-1}))$
  end if
  $s[i] \leftarrow \text{ContextNetwork}(cv)$ (Equations 1 to 4)
end for

Algorithm 4 dec_module

Input: predicted probability s, decision type dt, decision hyperparameter t/M, input word sequence x_{seq}, BoW input x_{BoW}
Return: $x^{\text{syn}}_{\text{BoW}}$ and $x^{\text{sem}}_{\text{BoW}}$
Initialize a zero decision vector $X \in \{0\}^V$.
for target words $w_i, \ldots, w_n$ **do**
  if dt= top_n **then**
    if $w_i \in top\_n(N, s[i])$ **then**
      $w_i^{cls} \leftarrow$ syntax
    else
      $w_i^{cls} \leftarrow$ content
    end if
  else if dt = probability_threshold **then**
    if $w_i \in prob\_threshold(t, s[i])$ **then**
      $w_i^{cls} \leftarrow$ syntax
    else
      $w_i^{cls} \leftarrow$ content
    end if
  end if
  index $\leftarrow$ index(w_i, vocabulary)
  if w_i^{cls} ==syntax **then**
    $X[\text{index}] \leftarrow X[\text{index}] + 1$
  else
    $X[\text{index}] \leftarrow X[\text{index}] - 1$
  end if
end for
$X_{\text{syn}} \leftarrow (X_{\text{syn}_1}, \ldots, X_{\text{syn}_V})$ where $X_{\text{syn}_i} = 1$ if $X_i > 0$ else $X_{\text{syn}_i} = 0 \; \forall i = 1, \ldots, V$
$X_{\text{sem}} \leftarrow (X_{\text{sem}1}, \ldots, X_{\text{sem}V})$ where $X_{\text{sem}i} = 1$ if $X_i \leq 0$ else $X_{\text{sem}i} = 0 \; \forall i = 1, \ldots, V$
$x^{syn}_{\text{BoW}} \leftarrow X_{\text{syn}} \cdot x_{\text{BoW}}$
$x^{sem}_{\text{BoW}} \leftarrow X_{\text{sem}} \cdot x_{\text{BoW}}$

target word, w_i, in the input sequence, where i ranges from one to n. For each target word, the algorithm retrieves context words, w_c, based on the specified context type, C_{type}, and context size, C_{size}. The algorithm then concatenates the embeddings of the context words, $E(w_c)$, based on whether the context type is symmetric or asymmetric. The concatenated embeddings, cv, are then passed through the context network model to obtain a prediction probabilities $s[i]$ for target word w_i. The process is repeated until all target words in the input sequence have been processed. The algorithm returns the prediction probabilities s at the end.

C Decision Module

Decision module is used to distinguish each word in syntactic or semantic class based on context network output.

We define Algorithm 4 which takes four inputs: the predicted probabilities s, the decision type dt, the decision hyper parameter t/M and the sequence of words x_{seq}. It returns two outputs: $x^{\text{syn}}_{\text{BoW}}$ and $x^{\text{sem}}_{\text{BoW}}$. The algorithm starts by initializing a zero decision vector, $X \in \{0\}^V$. It then iterates through each target word, w_i, for i ranging from one to n, the number of words in the document. Based on the decision type input, the algorithm checks whether the predicted probability for each target word falls within a certain range, either the top M values, or above a probability threshold. If the predicted probability falls within the specified range, the target word is classified as syntax, otherwise it is classified as content. The decision vector is updated accordingly, with syntax words having a positive value, and content words having a negative value. Syntactic (X_{syn}) and semantic (X_{sem}) binary decision vectors are then constructed based on positive and negative values of words in decision vector. The final output is calculated by taking the element-wise product of X_{syn} and X_{sem} with the BoW representation.

D Datasets and Preprocessing

For dataset preparation, we use SpaCy [Honnibal and Montani, 2017] to tokenize the data. We do not perform any additional pre-processing for SyConNTM. For other models, we remove common stop words, punctuations, and high and low frequency words. We determine high and low frequency words by removing the words which occur in more than 75% of the documents and in less than 50 documents.

E Model Details

Let x_{BoW} be the BoW representation of a document. The common encoder used across different models is:

Encoder:

$$
\begin{aligned}
X_1 &= \text{ReLU}(L^1 \cdot X_{\text{BoW}} + L^1_{\text{bias}}) \\
X_2 &= \text{dropout}(X_1, p_{\text{keep}}) \\
X_3 &= \text{BatchNorm}(L^2 \cdot X_2 + L^2_{\text{bias}}) \\
\text{enc\_out} &= \text{Softplus}(X_3)
\end{aligned}
$$

For D-VAE we use the following decoder with Dirichlet prior.

Decoder:

$$
\begin{aligned}
X_4 &= \text{BatchNorm}(L^3 \cdot z + L^3_{\text{bias}}) \\
\text{recon} &= \text{LogSoftmax}(X_4)
\end{aligned}
$$

For ETM we use the following decoder with both LogNormal and Dirichlet prior. α and δ represent topic and word embeddings respectively.

Table 1: Statistics for the datasets used in our experiments. Given is the average length of one document, the size of the vocabulary before preprocessing and after preprocessing.

	AVG. LENGTH	VOCAB	VOCAB (PREPROCESSED)
20NG	228	25,296	2,213
AMAZON REVIEWS	86	31,667	1,524
AG NEWS	44	25,736	203
IMDB REVIEWS	270	37,552	3,937
ROTTEN TOMATOES	23	6,124	159
YELP REVIEWS	152	30,065	2,690
GOV. REPORT SUMMARY	526	20,795	1,439

Decoder ETM:

$$X_4 = z \cdot \alpha \cdot \delta$$
$$\text{recon} = \text{LogSoftmax}(\text{BatchNorm}(X_4))$$

For SyConNTM we have two decoders, for syntax and semantics of the document.

Decoder SyConNTM:

Syntax:

$$X_4 = \text{BatchNorm}(L^3 \cdot z_{\text{syn}} + L^3_{\text{bias}})$$
$$\text{recon}_{\text{syn}} = \text{LogSoftmax}(\text{BatchNorm}(X_4))$$

Semantics:

$$X_5 = \text{BatchNorm}(L^4 \cdot z_{\text{sem}} + L^4_{bias})$$
$$\text{recon}_{\text{sem}} = \text{LogSoftmax}(\text{BatchNorm}(X_4))$$

F Ablation studies for SyConNTM

To decide on hyperparameter settings for the context network and decision module, we use the total number of stop words that are incorrectly identified as semantic words by the decision module. For this, we use the commonly-used stop word list made available by [Loper and Bird, 2002, NLTK] which contains 179 stop words. Throughout the experiments, we keep $\beta = 2$. All hyperparameters are given in Table 2.

F.1 Finding the optimal context size (c)

In order to determine the most appropriate context size, a series of experiments were conducted utilizing context sizes ranging from one to five for all the datasets. The context type and decision type were fixed at symmetric and top-5, respectively, throughout the experiment. The findings of the aforementioned experiment are presented in Figure 2. As can be observed, as the context size increases, there is a decrease in the number of stop words incorrectly identified as semantic words. Hence, for the rest of experiments, we select context size of five.

F.2 Finding the optimal context type (C_{type})

In order to evaluate the performance of different context types, a series of experiments were conducted utilizing symmetric and asymmetric context types. The results, shown in Figure 3, indicate that the symmetric context type performs significantly better across all datasets. The decision type was fixed at top-5 and the context size was kept constant at 5 throughout the experiment. Based on these results, the symmetric context type was selected.

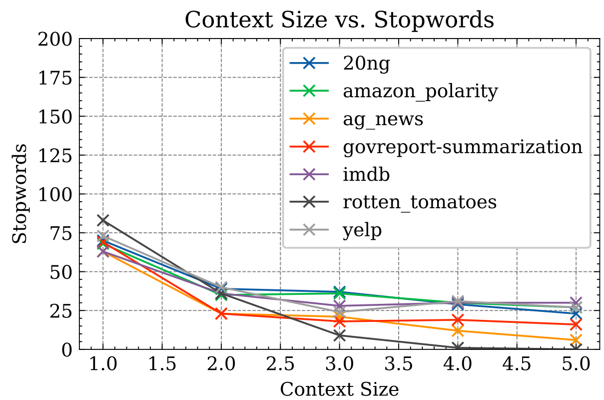

Figure 2: Finding the optimal context size. As the context size increases, there is a decrease in the total number of stop words incorrectly classified as semantic words.

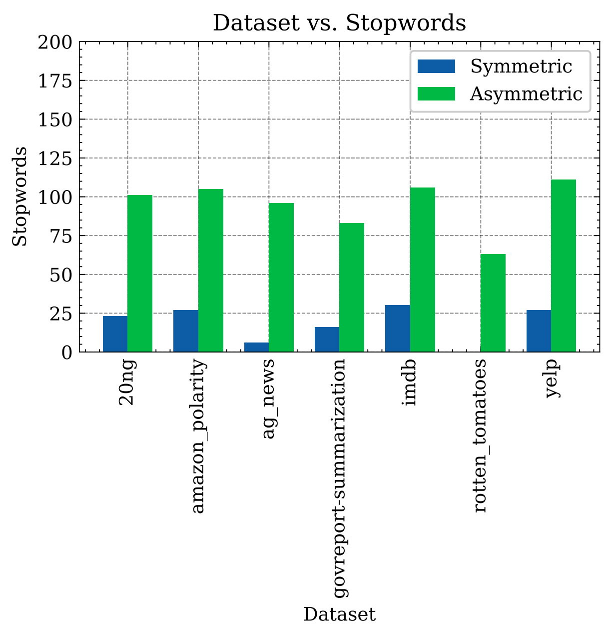

Figure 3: Finding the optimal context type. The results demonstrate that the symmetric context type consistently performs better across all datasets, with a lower number of stop words incorrectly classified as semantic words.

Table 2: Hyperparameter settings for the experiments.

BATCH SIZE	128
TRAIN:VAL:TEST	70:15:15
α	0.02
λ	0.5
β	2
LEARNING RATE	0.001
MAX EPOCHS	100
CONTEXT SIZE	5
CONTEXT TYPE	SYMMETRIC
DECISION TYPE	TOP-M
M	5
L^1	$\mathbb{R}^{\text{VOCAB\_SIZE} \times 500}$
L^2	$\mathbb{R}^{500 \times \text{Z}}$
L^3	$\mathbb{R}^{\text{Z}_{syn} \times \text{VOCAB\_SIZE}}$
L^4	$\mathbb{R}^{\text{Z}_{sem} \times \text{VOCAB\_SIZE}}$
p_{keep}	0.25
WORD EMBEDDINGS	[PENNINGTON *et al.*, 2014, GLOVE]

F.3 Finding the optimal Top-M/Probability Threshold

In order to evaluate the performance of different decision types, a series of experiments were conducted utilizing top-m ($M = 1, 3, 5$) and probability threshold ($t = 0.1, 0.3, 0.5$) decision types. The results, as shown in Figure 4, indicate that top-m decision type generally performs better, with top-5 performing the best across all datasets. The context type was fixed at symmetric and the context size was kept constant at five throughout the experiment. Based on these results, the top-5 decision type was selected for experiments.

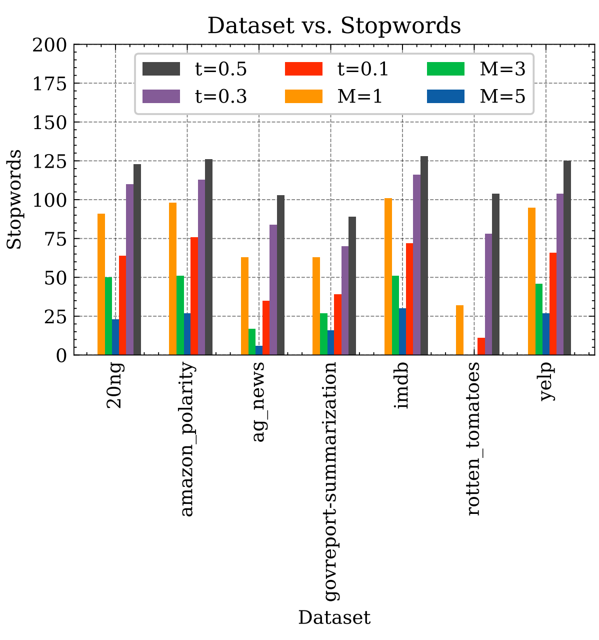

Figure 4: Finding the optimal decision type. The results demonstrate that the top-m decision type generally performs better, with top-5 decision type achieving the highest level of accuracy across all datasets.

G Syntax and Semantic Topics

G.1 Yelp:

Semantic Topics

- casino pool lounge club clubs dj bathrooms crowded security bathroom
- poke vermicelli viet smoothie ra regulars asians gyoza juices ordinary
- lobster creamy ravioli brunch gnocchi coffee crepes pastry desserts dinner
- stores museum shopping products gear jewelry books target changing classes
- excessive quicker glorified laces borderline helpful lingering patron musky pa
- pour oder manger frugal slip roughly shocking largely thier hubs
- thier kings utterly overlooked quicker flash struggle remained snap anthony
- pizza wings buffalo burgers gyro pita toppings combo joint sandwiches
- place & great maybe sushi try still water chef fish
- business today work customers issues lady needed office delivery situation

Syntactic Topics

- rude gone bad worse charge drive anyone received hair done
- enjoyed great delicious amazing awesome excellent dry thai pork salsa
- un le et les pas est la en du de
- fabled sidney yellowtail revoir instantly bleck spicy swim experiece hiring
- . , ( ) ” _ * –
- i me my she in and have in . for

- always has strip area can vegas selection places may feel
- were we was with like but good of it and
- und die ich der ist es das nicht war kann
- said she back when we after asked service time us

LDA Topics

- . the , a and to you is of it
- i . was and the my to she a me
- . and the to they a for , service we
- . i to the ! a for it $ and
- . the , and a was i of it to
- . the , i and was it a to of
- we . the to and was our , he us
- the . , and was to a in room i
- i , . the to and a of my that
- ! . and the is i , great this place

LDA Topics (preprocessed)

- sushi buffet place food fish good roll rice rolls eat
- pizza good sauce like ice cream place cheese sweet tea
- like time place store know new going car good work
- food great place service good bar time staff beer love
- room hotel like vegas place night nice stay people good
- minutes service said told order time asked got manager went
- food good place burger fries like better sandwich pretty meat
- ordered food like came table restaurant server got waitress meal
- coffee love place amazing best delicious menu breakfast great wine
- good chicken salad lunch food great service steak soup nice

D-VAE Topics

- , of the to . and - i as with
- me i she they told to said my out do
- et un pour le de les la pas en est
- vegas most are has want its your those area mall
- worst horrible waited waiting walked attitude arrived paid min left
- delicious sandwich cheese thai sauce rolls special tacos salad fries
- awkward besides parmagiana themselves vodka reordering ditching succeeding closer bulging
- food restaurant was with (were our cheese us it
- und die das der m ist nicht war zu es
- highly beautiful thank color helped beyond wedding thanks treat recommend

D-VAE Topics (preprocessed)

- dish try sauce ordered soup meat chicken fried better cream
- quick kitchen oh fan ordering mexican seating style gets kept
- hotel room vegas rooms told free called stay stayed said
- food minutes restaurant bar service order drinks ordered waiter waitress
- amazing great love awesome staff try lunch excellent attentive family
- bar place great night happy sushi good menu food drinks
- food place service good better great price sushi love buffet
- fries burger place sandwich try meat love ordered good food
- place bar find love music looking club store vegas people
- told said service order called car know store time manager

ETM-D Topics

- customer called never him guy money business car call his
- ja selbst dieser wurde meinem anderen bedienung abend stunden zimmer
- the . this on a and in that i not
- fait avec leur suis toujours leurs du ou en faire
- was were our so had but we table . at
- sips creators goodyear firday nibbling fake carrying elm rusty ginza
- fake creators elm recognize sips rusty firday bat girlies exacting
-) but all (as this like a $ with
- fries plate salsa served soup fish sweet pho french delicious
- free always walk has parking floor looking most area enjoy

ETM-D (preprocessed)

- rice sushi chicken like food fried roll good buffet fish
- wine great time happy table menu dinner server bar night
- good place pizza food great bar selection love better prices
- room hotel rooms pool stay floor check strip night vegas
- car phone told called customer work time said problem business
- minutes food order said waitress asked table took waited drink
- love look fun kids store staff work new shop feel
- pizza cheese sandwich good sauce salad fries bread ordered flavor

- friendly staff atmosphere great attentive excellent fantastic clean recommend highly
- expected totally quickly available read mediocre decor believe expensive oh

G.2 20ng:

Semantic Topics

- serious finding behavior basically riding sensitive definitely story yourself funny
- build standard & mark phi hardware mon using text full
- monitor setup ps resolution zero dealer fits feature price installed
- build just already service before us only without called because
- christian man religion us jewish because evil without yourself guns
- riding measure kept water funny sensitive definitely possibly obviously weight
- annual skylab nagorno ti badly inevitable orteig dizziness endowment sspx
- xbiff annual skylab nagorno ti badly inevitable orteig dizziness endowment
- richter annual skylab nagorno badly ti inevitable orteig dizziness sspx
- bhj richter ghj gallant ymon apda kodiaks pena xbiff claris

Syntactic Topics

- _ < = ′ : > − ; ∗)
- point away believe were between them someone world time too
- push papers armed announced rockets contest stars genocide heads et
- this my as that would not but no or time
- published nuclear vote contest generation committee goals penalty powerful guidelines
- controller windows bought uses worked drives box anybody reply large
- george scientific authors contrib north shuttle round soviet ice protect
- entries programs important rates latest postscript solution couple contact digital
- and . that of this is in the to ,
- asking quality york won kings ii st sale byte toronto

LDA Topics

- * .) (the , - to and a
- . , the of - and a to : (
- # . the , of is it to and
- > ' / . x ; , : - the
- . the , to a i and is it of
- the , . of to and in a + 0.010*
- = - < : , (. >) +
- . the , of + 0.004* + 0.004* + 0.004* + 0.004* + 0.003*
- _ . the , - i to a and db
- . , the to that of + 0.020* + 0.018* + 0.018*

LDA Topics (preprocessed)

- game window better good play got win team like best
- people think way know good law government believe human life
- know think like said going mr lot people things time
- space new center university war number april list research press
- use like sure work word system people program available know
- want time people like know going good left new cost
- new card drive disk car video price problem like available
- file available key use data code bit files chip like
- years ago group like center year anybody weeks day believe
- people god think government time believe like year right said

D-VAE Topics

- ' _ > - = (: / <].txt
- never same see good very does really now look might
- senior master color details members press usa jobs league commercial
- was this it are who of " as in they
- increased tape vote hockey myers ahead satellite lock director papers
- never same look law effect let post case enough number
- authors bill sj south stars received flame postings nuclear picture
- . the of are this in (a - it
- sale shipping asking please sell excellent advance condition stereo cd
- division rocket papers premises magazine press mission document myers reported

D-VAE (preprocessed)

- thanks help bit memory problem anybody disk work drive windows
- thought took home subject taken problems place having agree response
- body force questions turn change contact effect longer buy able
- key bit probably chip think know work problem system trying
- good took second home year car great years best bad

- bit research law university data key graphics public software program
- god law people rights second jesus new human gun issue
- space new people know work years university said let list
- version file graphics dos window set software windows bit run
- god know jesus bible think people said believe body man

ETM-D Topics

- menial istanbul invoke cliff date metacard scars press billings rose
- do better than must us might point its true make
- do was we _ i he it you not they
- > * - = : ’ <) (;
- menial measure istanbul papers quest trade metacard thus cliff zephyr
- the . , a -) in and is on
- date menial measure round devices flight attitude quest papers separate
- getting better drive controller must chip help mac maybe problems
- character menial washington istanbul english trade track scars cliff echo
- menial scientists south cliff satellites release waive wing workspace quest

ETM-D Topics (preprocessed)

- people god jesus christian bible believe know right said like
- games good bad team game come year went car years
- key bit chip system data number public use access government
- dos video advance set thanks running write speed run windows
- april david lost center article games city week deal interested
- drive dos advance thanks card windows anybody disk pc key
- drive windows file use program pc like software window info
- people government states law program public university years year state
- people government public years key right new power law space
- argument believe come jesus evidence certainly bible christian true body

H Motivation from Cognitive Science and Linguistics

Cognitive science and linguistic research suggest that words appear in sentences for two reasons: to fulfill a syntactic function or to convey semantic content. Studies have shown that syntax and content words elicit different patterns of brain activity [Neville *et al.*, 1992] and have distinct developmental tendencies [Brown, 1973]. Additionally, it has been reported that humans acquiring language infer syntactical categories of words based on their short-term context within a document, while content words rely on long-range context [Redington *et al.*, 1998].

Theoretical linguistic considerations and psycholinguistic studies have shown that grammatical functions (*syntax*) and semantics (*content*) are distinct subprocesses within the language domain [Neville *et al.*, 1992]. Syntax and content - both terms refer to a set of words that help the writer create correct sentences that the reader understands. In natural language, syntax refers to the grammatical rules that determine the structure of the sentence. Semantics, on the other hand, deals with the content of a text. Sometimes grammatically correct words do not make sense even though they are grammatically sound. Semantics helps add a layer of meaning so that words together make sense [Culicover and Jackendoff, 2006].

A sentence in a corpus is a complex sequence of words, and the context determines the meaning of each word. For example, the word “sweet” can mean a flavor opposite of sour or something amazing, depending on the context. Similarly, syntax rules are also governed by the context in which they are used. Thus, syntax and semantics are rarely context-free because the surrounding words in a sentence or paragraph directly influence both grammatical rules and the meaning [Friederici and Weissenborn, 2007].

As for context, syntactic words have a short-range context in a document, while semantic words have a long-range context [Griffiths *et al.*, 2004]. Syntactic constraints lead to relatively short-range dependencies that span multiple words within a sentence. Semantic conditions lead to long-range dependencies: different sentences in the same document are likely to have similar content and use similar words. Hence the distinction between syntax and content words in a corpus is context-dependent. The context of words and interpretation is often subjective, leading to additional complications in defining syntax and context words in a corpus.

In tasks such as topic modeling and text classification that depend on the content of a corpus, syntax words are usually removed [Blei *et al.*, 2003]. And since the distinction between syntax and content words is not so clear, one is often limited to using manually curated word lists. Syntax word lists usually consist of stop words and punctuation. They are also task-specific, and sometimes a higher frequency of word occurrence is also considered as syntax [Griffiths *et al.*, 2004].

Curating a list of universal syntactic and semantic words is inefficient because their context is local and corpus dependent. At the local level, it is also possible for a word to have both features in a different context [Griffiths *et al.*, 2004]. Because these lists are not uniform, model performance and benchmarking are also inconsistent [Hoyle *et al.*, 2021].

Regarding linguistic composition, this work aims to automate the distinction of syntax and content words based on context.

I Limitations

Few limitations of the proposed work are:

- In our framework, we focus on learning topics from syntactic words and ignore syntactic classes. Since syntactic classes are beneficial for determining part-of-speech (POS) tags, in future developments, we intend to incorporate the learning of both syntactic classes and topics from syntactic words.
- The scope of this research is centered on the integration of syntax with neural topic models, and does not specifically address the investigation of syntactic topics or the determination of an optimal number of syntactic topics to be learned within our framework.

J Things we tried that did not work

In efforts to combine syntax and neural topic models, we tried several architectures, but only three were found to be effective, albeit not as superior as the proposed architecture. The purpose of this section is to present these architectures and provide a brief overview so as to aid future research in this field. The architectures are presented in Figure 5, 6, and 7.

Syntax aware attention GAN: In this study, two adversarial generators were implemented to generate the syntactic and semantic components of a document corpus, which were subsequently added to and compared against the original document using a discriminator as shown in Figure 5. To supervise the model with syntactic and semantic representations, a sequential network incorporating self-attention was employed. The self-attention matrix and attention scores were utilized to identify words with both long-range and short-range attention. A significant challenge encountered with this model is that, when utilizing a transformer within the sequential model, it operates efficiently only with sub-word tokenizers rather than word-level tokenizers, which are crucial for identifying topics. Additionally, alternative sequential networks such as Recurrent Neural Networks (RNNs) require processing a significant amount of text (in terms of length) to accurately determine attention scores. Furthermore, determining an appropriate threshold for identifying both long-range and short-range context through attention scores remains a problem.

A separate Decision Layer: In this model, a decision layer was introduced following the encoder in order to determine the allocation of each word in a document to either syntax or semantics as shown in Figure 6. As the decoders in this model do not have separate supervision, the similarity of the outputs from the decoders was maximized in order to enforce their differentiation. The resulting outputs were then combined to form the predicted representation of the document. One limitation of this model is that the separation between syntax and semantic words in the document corpus was not always clear-cut.

A seq-net topic model: In this model, which is similar to the final proposed model, a sequential network was utilized to learn the class of each word as either syntax or semantics in an unsupervised manner as shown in Figure 7. The objective was the loss of the VAE model. While the model was able to differentiate syntactic words to some extent, it was unreliable and unclear why it worked for some datasets and not for others.

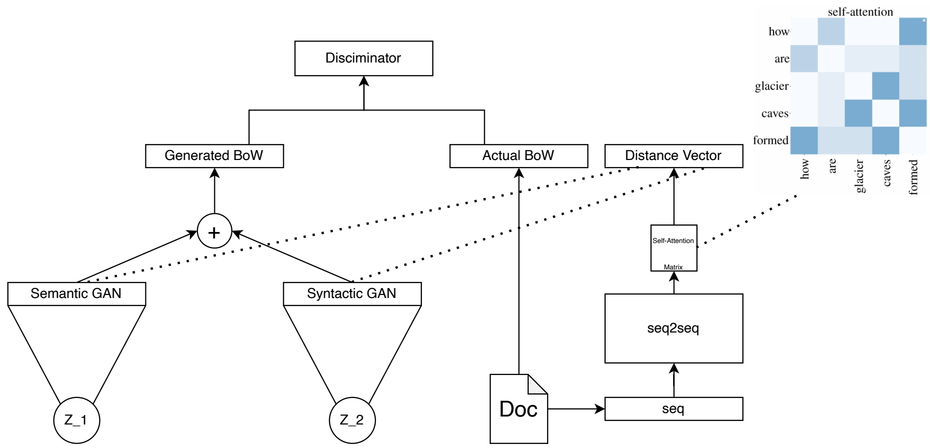

Figure 5: Syntax aware attention GAN - This figure illustrates the implementation of two adversarial generators to generate the syntactic and semantic components of a document corpus, which are then added to and compared against the original document using a discriminator. The separation is guided by the attention scores from a self-attention matrix.

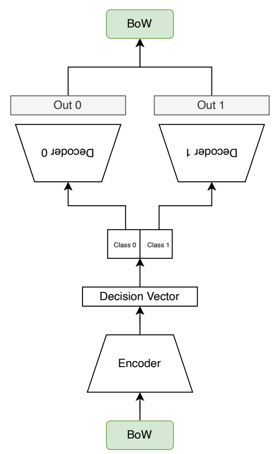

Figure 6: A separate Decision Layer - This figure illustrates the implementation of a decision layer following the encoder to determine the allocation of each word in a document to either syntax or semantics.

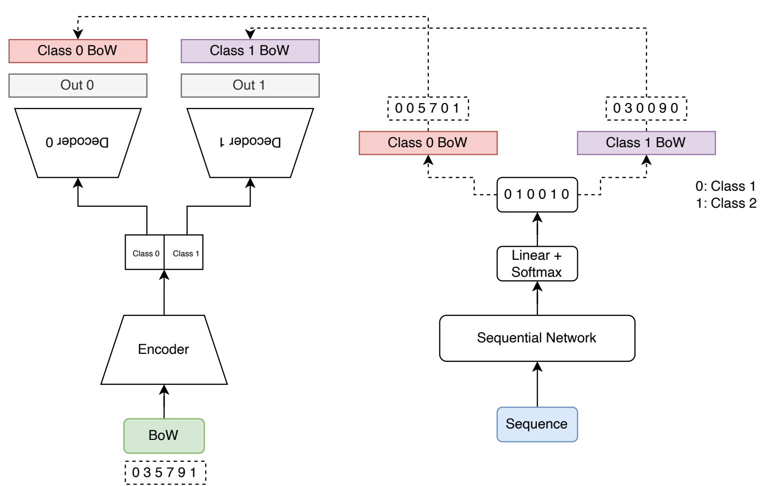

Figure 7: A seq-net topic model - This figure illustrates the implementation of a sequential network to classify each word as either syntax or semantics in an unsupervised manner using a sequential network.

References

[Blei *et al.*, 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[Brown, 1973] R Brown. A first language (pp. 54). *Cambridge, MA: Harvard University Press*, 1973.

[Chelba *et al.*, 2017] Ciprian Chelba, Mohammad Norouzi, and Samy Bengio. N-gram language modeling using recurrent neural network estimation. *arXiv preprint arXiv:1703.10724*, 2017.

[Culicover and Jackendoff, 2006] Peter W Culicover and Ray Jackendoff. The simpler syntax hypothesis. *Trends in cognitive sciences*, 10(9):413–418, 2006.

[Friederici and Weissenborn, 2007] Angela D Friederici and Jürgen Weissenborn. Mapping sentence form onto meaning: The syntax–semantic interface. *Brain research*, 1146:50–58, 2007.

[Griffiths *et al.*, 2004] Thomas Griffiths, Mark Steyvers, David Blei, and Joshua Tenenbaum. Integrating topics and syntax. *Advances in neural information processing systems*, 17, 2004.

[Honnibal and Montani, 2017] Matthew Honnibal and Ines Montani. spaCy: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[Hoyle *et al.*, 2021] Alexander Hoyle, Pranav Goel, Andrew Hian-Cheong, Denis Peskov, Jordan Boyd-Graber, and Philip Resnik. Is automated topic model evaluation broken? the incoherence of coherence. *Advances in Neural Information Processing Systems*, 34:2018–2033, 2021.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456. PMLR, 2015.

[Loper and Bird, 2002] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[Neville *et al.*, 1992] Helen J Neville, Debra L Mills, and Donald S Lawson. Fractionating language: Different neural subsystems with different sensitive periods. *Cerebral Cortex*, 2(3):244–258, 1992.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on EMNLP (EMNLP)*, pages 1532–1543, 2014.

[Redington *et al.*, 1998] Martin Redington, Nick Chater, and Steven Finch. Distributional information: A powerful cue for acquiring syntactic categories. *Cognitive science*, 22(4):425–469, 1998.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.