Molecular Translation: Interpreting Organic Compound Images with Image Captioning

Boyuan Liu

Georgia Institute of Technology
bryanliu@gatech.edu

Abstract

Organic compounds are prevalent in the fields of chemistry and biology. They are often represented in diagrams of skeletal formulas to give comprehensive overviews of their structures. Searching for organic compounds in the diagrams in literature is not an easy task and often requires inefficient manual work. In this paper, we illustrated a versatile image captioning pipeline that can adapt to different tasks by switching the models in both the encoder and decoder. We experimented with EfficientNet, MobileNet, ResNet, Vision Transformer, and Transformer in Transformer for the encoder. We also experimented with Long Short-Term Memory (LSTM) and Transformer for the decoder. The EfficientNet B2 encoder + Transformer decoder structure had the best overall performance among the different setups tested in the experiments.

1. Introduction

The advances in image captioning in the past few years have made many previously unachievable tasks possible. Researchers went from describing dogs in images to assisting visually impaired people [1][2][3] [4]. In the field of organic chemistry, chemists have been using the skeletal formula, a structural notation of chemical compounds, for centuries. These image representations provide intuitive overviews of the chemicals, but they are very challenging to search for in the literature. Finding the exact organic compound often requires inefficient manual work since there are estimated to be nearly 24 million organic compounds [5]. Researchers often find themselves digging through piles of documentation, searching for information related to an organic compound. Image captioning is a supervised learning task that uses both computer vision and natural language processing techniques to generate textual description of an image. Automated recognition of optical chemical structures via image captioning would help researchers speed up this process significantly. Previous research have used methods such as the rule-based OSRA [6], an approach based on OpenNMT [7], and a feature extraction-assembly approach called OCR [8]. These methods have various limitations, such as accuracy, the complexity of the organic compounds, or image quality requirements.

In this paper, we implemented the widely adopted encoder-decoder structure in image captioning and experimented with various state-of-the-art image models to find the best architecture for this task that balanced efficiency and performance. The dataset used in this paper contains approximately four million images of organic compounds synthetically generated by Bristol-Myers Squibb (BMS) [9]. These images vary in rotation angles, resolutions, and noise levels. This paper aims to build an efficient image captioning pipeline that can process batches of organic compound images and accurately output their International Chemical Identifier (InChI) [10] text representation.

2. Related Work

OSRA[11]: It is an open-source chemical structure recognition tool developed by Igor Felippov[11]. It is a utility designed to translate raster image based representation of chemical structures, as they appear in scientific journals, patent documents etc. into a computer chemical structure format. OSRA can generate the SMILES notations or MOLfile of the molecular structure images encountered within that

document. It is written in C++ and uses CImg [12] library for image processing and GOCR[13] for text recognition. The computation of nxn distance matrix of connected components, which is used for page segmentation in OSRA, makes the entire processing very slow if a page has huge paragraphs of text as compared to a page having only structures.

CLiDE[12]: It is a chemical OCR tool aimed at automated extraction of chemical information from either electronic PDF version or from the printed chemistry literature. CLiDE development started back in 2008 and it has been commercially available in three versions: 1) CLiDE Standard (one structure recognition per process), 2) CLiDE Professional (a large number of chemical structures per process), and 3) CLiDE Batch (supports batch processing and web service environments). CLiDE needs document images of at least 300 dpi resolution to achieve reliable result. However, the embedded molecular diagrams in document usually have 72–96 dpi. Due to license requirements caveat and the tool being not open-source, we could not sort out which Image processing and OCR engines were used in CLiDE. Moreover, we were also not able to test the data sets on this tool, therefore, the evaluation section only includes comparison with OSRA. Standards for Representation (SMILES and MOL): SMILES and MOL are the chemical formats that contains information about a chemical structure that is machine readable. This processed data can be stored in databases for global usage across various disciplines.

3. The Bristol-Myers Squibb Dataset

The dataset used in this paper was synthetically generated by Bristol-Myers Squibb [9]. It contains approximately four million images of organic compounds. The images in the dataset have various resolutions, noise levels, and rotation angles. The labels are the InChI text representation of the corresponding organic compound. Figure 1 shows an example from the dataset. The examples have corruption in various regions. The image contains letters N, H, O and F, representing nitrogen, hydrogen, oxygen, and fluorine atoms, '—' and '==' representing the single and double bonds between atoms, and '⌬' representing benzene rings. The organic compound shown in Figure 1 has InChI label lengths of 137 and 197, respectively. The examples in this dataset have label lengths that range from 23 to 277.

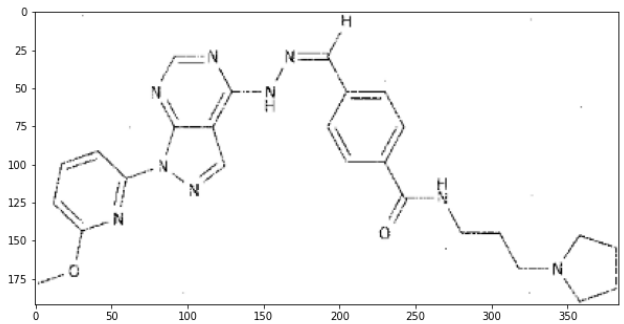


Figure 1 - Training examples from the dataset. Letters N, H, O and F represent nitrogen, hydrogen, oxygen, and fluorine atoms, '—' and '==' represent the single and double bonds between atoms, and the '⌬' symbol represents a benzene ring.

4. Image Captioning Architecture

Image captioning utilizes a combination of computer vision and natural language processing techniques to predict captions based on visual inputs. The multimodal log-bilinear model introduced by Kiros et al. [19] was the first attempt to use neural networks for image caption generation. Mao et al. first used a recurrent neural network instead of a feed-forward model [20]. The widely cited *Show, Attend and Tell* by Xu, Kelvin, et al. [21] introduced a CNN encoder + LSTM with attention mechanism decoder architecture. The CNN encoder extract feature information from the images, and the LSTM decoder generates text sequences based on the extracted features. As a result, the model was able to learn alignments that corresponded very strongly with human intuition and achieved state-of-the-art performance on the Flickr8k, Flickr30k, and MS COCO benchmark datasets [21].

Image captioning allows us to automate the task of generating text representations for any image, which makes it very suitable for the molecular translation task. In this paper, we built an image captioning pipeline that used a preprocessing-encoder-decoder architecture, similar to the architecture used by Xu, Kelvin, et al. [21]. To make the image captioning pipeline more versatile, we designed the pipeline in a way that allows easy switches among models. We implemented EfficientNet, MobileNet, ResNet, Vision Transformer, and Transformer in Transformer for the encoder and LSTM/Transformer for the decoder. This design provides flexibility and makes it easier to search for the best models for a given task.

4.1 CNN/Vision Transformer Encoder

The image captioning pipeline has an encoder block that can adopt a variety of CNN or Vision Transformer models. The encoder takes preprocessed organic compound images as inputs and performs feature extraction to generate embedded feature vectors that will be passed as input to the decoder. Some of the implemented models are EfficientNet, MobileNet, ResNet, Vision Transformer, and Transformer in Transformer.

The EfficientNet was the primary model used in the molecular translation task. EfficientNet is a state-of-the-art computer vision architecture. It uses a compound scaling technique to uniformly scales model width, depth, and resolution in a principled way [17] with a compound coefficient φ. The scaling factors are calculated using the equations below, while α, β, γ specify how much extra resources are assigned to network width, depth, and resolution respectively. EfficientNet B0 to B5 were tested and compared to the other vision models in the experiments.

$$\begin{aligned} \text{depth: } & d = \alpha^\phi \\ \text{width: } & w = \beta^\phi \\ \text{resolution: } & r = \gamma^\phi \\ \text{s.t. } & \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\ & \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{aligned}$$

4.2 Transformer/LSTM Decoder

Similar to the encoder, the decoder can choose between LSTM and Transformer models. For the LSTM model, we leveraged a two-layer LSTM with attention mechanism that closely follows the one used in Zaremba et al [21][22]. The feature vectors from the encoder are fed in as input to the attention and LSTM layers, which generate a predicted text sequence. LSTM layers are defined by the following equations, where, i_t, o_t, f_t, c_t, h_t are the input, output, forget, memory and hidden state respectively. The vector $\hat{\mathbf{z}}$ is the context vector and $\mathbf{E}$ is the embedding matrix [21]. We chose this architecture to take advantage of LSTM's ability to counter the vanishing gradient problem as the sequence length increases. The attention mechanism allows the decoder to attend to all the hidden states from the encoder and focus on certain parts of the input to better understand its context.

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}_t} \end{pmatrix}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t).$$

The Transformer model further utilizes the attention mechanism. Instead of using a single attention layer before the LSTM layers, the Transformer model can stack several multi-head attention blocks together. The self-attention blocks use the following equation to compute the dot product of the query with all keys, scaled by the input dimension, and apply a softmax function to obtain the weights for the values [18]. This weighted values can be viewed as values with added context information. This ultimately leads to better predicted sequences and faster training.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Figure 2 shows the CNN + Transformer architecture. The CNN model takes preprocessed images and extracts features that are later fed into the Transformer encoder. The outputs from the Transformer encoder are used as keys and values by the decoder blocks to generate sequences.

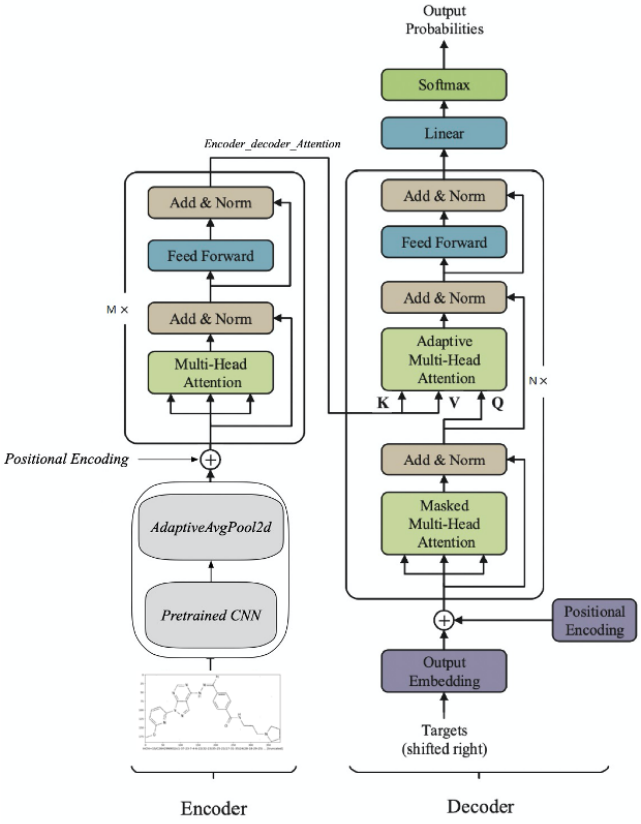


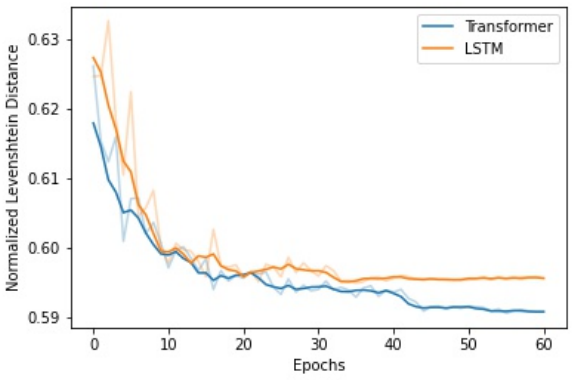
Figure 2 - The CNN + Transformer Architecture [18]. The CNN model takes preprocessed images and extracts features that are later fed into the Transformer encoder. The outputs from the Transformer encoder are used as keys and values by the decoder blocks to generate sequences.

5. Results and Discussion

Several experiments were conducted to find the optimal encoder/decoder setup for the E2E pipeline. Due to the large size of the training set, these experiments used a randomly sampled subset of 10,000 images for training and validation. The models were trained with the full dataset after the optimal setup was determined.

5.1 Transformer v.s. LSTM

We adopted the LSTM with attention architecture introduced by Xu, Kelvin, et al. [21] and implemented a Transformer decoder as an alternative option. To compare the performance of LSTM and Transformer decoders, we used a pre-trained EfficientNet B0 as the encoder and trained both decoders using the same hyperparameters. We used the Levenshtein distance [23] and accuracy as the primary metrics to evaluate the model's performance. The validation accuracy and normalized Levenshtein distance are shown in Figure 3. We used the moving average with a sliding window of 5 epochs to better show the overall trend. The Transformer decoder had higher accuracy and lower Levenshtein distance after 60 epochs of training. The LSTM decoder also appears to converge sooner at 20 epochs, while the Transformer decoder's accuracy was still improving after 60 epochs. The EfficientNet + LSTM setup took 240 seconds per epoch on average. On average, the EfficientNet + Transformer setup took 142 seconds per epoch, which is 40% faster. The Transformer decoder had better performance and faster training time. Therefore, the Transformer is the better decoder option for the molecular translation task.



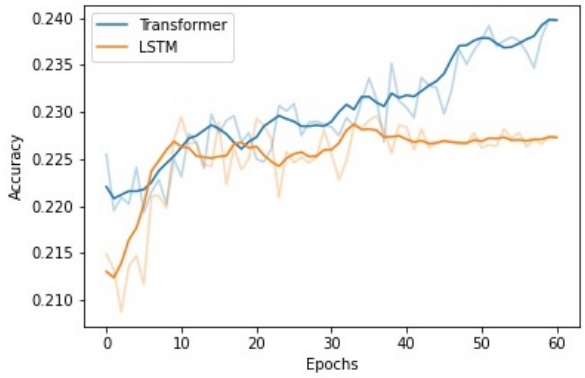


Figure 3 - Validation Accuracy and Normalized Levenshtein Distance of LSTM and Transformer. The Transformer decoder had higher accuracy and lower Levenshtein distance after 60 epochs. The EfficientNet + Transformer setup was also 40% faster than the EfficientNet + LSTM setup.

5.2 Encoder Vision Model Comparison

After choosing the Transformer model as the decoder, we used the same method to compare several state-of-the-art models for the encoder, including EfficientNet, MobileNet, ResNet, Vision Transformer (ViT), and Transformer in Transformer (TNT). These encoder models were paired with the same Transformer decoder and trained using the same hyperparameters. Figure 4 shows the normalized Levenshtein distance and accuracy during validation. We used the moving average with a sliding window of 5 epochs to better show the overall trend. EfficientNet B0 and B2 outperform the other models by a prominent margin in terms of both validation accuracy and Levenshtein distance. EfficientNet B2 had the third-best training time of 178 seconds per epoch on average, closely behind EfficientNet B0 and MobileNet. The details of the models' performance after 60 epochs of training are shown in Table 1. The EfficientNet B2 + Transformer structure was chosen eventually after balancing performance and training time.

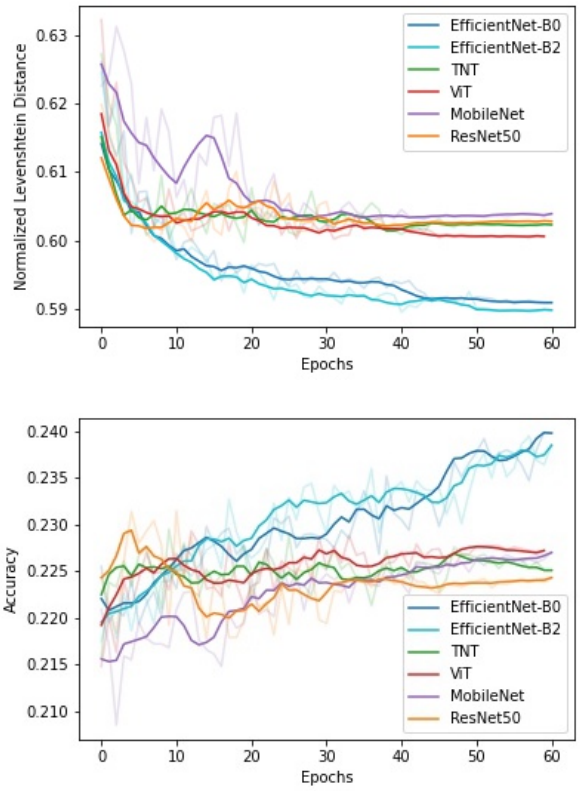


Figure 4 - Validation Accuracy and Normalized Levenshtein Distance of Encoder Models. EfficientNet B2 outperforms the other models by a prominent margin while having the third-best training time, closely behind EfficientNet B0 and MobileNet.

	Normalized Levenshtein	Accuracy	Time (s) / Epoch
EfficientNet B2	0.5898	0.2385	178
EfficientNet B0	0.5909	0.2398	142
ViT	0.6006	0.2272	253
TNT	0.6023	0.2251	261
ResNet50	0.6028	0.2243	180
MobileNet	0.6039	0.2270	159

Table 1 - Encoder models' performance after 60 epochs of training on the 10,000 images subset (ranked by their normalized Levenshtein distances).

5.3 Final Results

Based on the results from the previous experiments, the EfficientNet B2 encoder + Transformer decoder structure has the best overall performance on the 10,000 images subset. After hyperparameter tuning, this setup was trained with the full 4 million images dataset for 45 epochs. The validation results in Figure 5 show that the training accuracy reached 99.7% at the end of the training, while the validation accuracy was lower by a small margin. This indicates that the model experienced a certain level of overfitting. The model reached high accuracy and low Levenshtein distance after ten epochs and continued improving until it converged at 40 epochs. The model eventually achieved 96.9% accuracy and 1.88 Levenshtein distance in validation and 3.58 Levenshtein distance in testing. It performed better than some of the approaches in previous research, such as OSCR and CSR [8], by a noticeable margin.

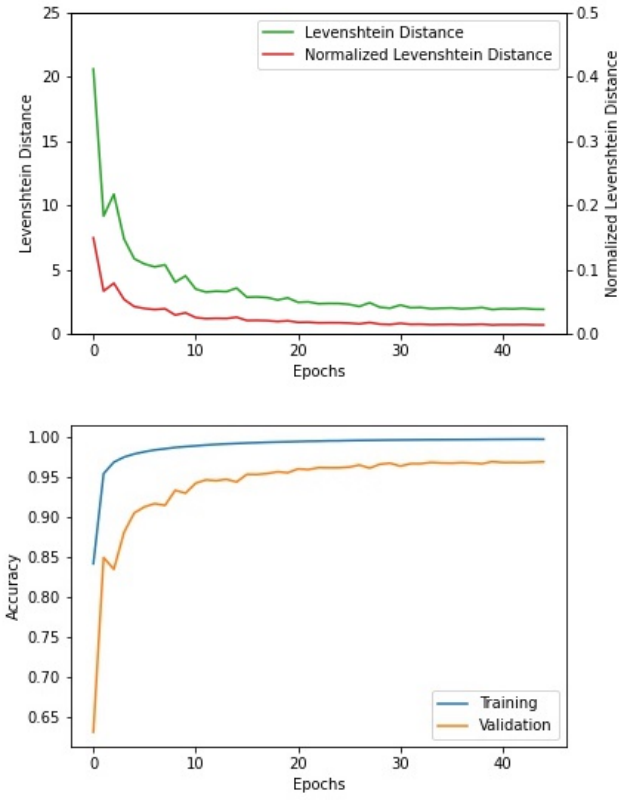


Figure 5 - Accuracy and Levenshtein Distance of the Final Model. The model eventually achieved 96.9% in accuracy and 1.88 in Levenshtein distance after converging at 40 epochs.

6. Conclusion

In this paper, we built an end-to-end image captioning pipeline for the molecular translation task. The image captioning pipeline incorporated different models such as EfficientNet, ResNet, Transformer, and LSTM for both the encoder and the decoder. After the experiments, the EfficientNet B2 encoder + Transformer decoder structure proved to be the optimal setup for the molecular translation task. The model eventually achieved 96.9% accuracy and 1.88 Levenshtein distance in predicted sequences. This image captioning pipeline's ability to easily switch among models for both encoder and decoder provides excellent versatility, which can help the pipeline adapt to other image captioning tasks.

References

[1] Makav, Burak, and Volkan Kılıç. "A new image captioning approach for visually impaired people." *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*. IEEE, 2019.

[2] Makav, Burak, and Volkan Kılıç. "Smartphone-based image captioning for visually and hearing impaired." *2019 11th international conference on electrical and electronics engineering (ELECO)*. IEEE, 2019.

[3] Rane, Chinmayi, et al. "Image Captioning based Smart Navigation System for Visually Impaired." *2021 International Conference on Communication information and Computing Technology (ICCICT)*. IEEE, 2021.

[4] Sidorov, Oleksii, et al. "Textcaps: a dataset for image captioning with reading comprehension." *European conference on computer vision*. Springer, Cham, 2020.

[5] Lipkus, Alan H., et al. "Structural diversity of organic chemistry. A scaffold analysis of the CAS Registry." *The Journal of organic chemistry* 73.12 (2008): 4443-4451.

[6] Filippov, Igor V., and Marc C. Nicklaus. "Optical structure recognition software to recover chemical information: OSRA, an open source solution." (2009): 740-743.

[7] Tabchouri, Sophia. A machine learning approach to molecular structure recognition in chemical

literature. Diss. Massachusetts Institute of Technology, 2019.
[8] Bukhari, Syed Saqib, Zaryab Iftikhar, and Andreas Dengel. "Chemical Structure Recognition (CSR) System: Automatic Analysis of 2D Chemical Structures in Document Images." *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019.
[9] BMS-Molecular-Translation. 2021. Available online: https://www.kaggle.com/c/bms-molecular-translation
[10] Heller, Stephen R., et al. "InChI, the IUPAC international chemical identifier." *Journal of cheminformatics* 7.1 (2015): 1-34.
[11] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[12] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
[13] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
[14] Howard, A. G., et al. "Efficient convolutional neural networks for mobile vision." *arXiv preprint arXiv:1704.04861* (2017).
[15] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
[16] Huang, Yanping, et al. "Gpipe: Efficient training of giant neural networks using pipeline parallelism." *Advances in neural information processing systems* 32 (2019).
[17] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International conference on machine learning*. PMLR, 2019.
[18] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
[19] Kiros, Ryan, Ruslan Salakhutdinov, and Rich Zemel. "Multimodal neural language models." *International conference on machine learning*. PMLR, 2014.
[20] Mao, Junhua, et al. "Deep captioning with multimodal recurrent neural networks (m-rnn)." *arXiv preprint arXiv:1412.6632* (2014).
[21] Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." *International conference on machine learning*. PMLR, 2015.
[22] Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization." *arXiv preprint arXiv:1409.2329* (2014).
[23] Yujian, Li, and Liu Bo. "A normalized Levenshtein distance metric." *IEEE transactions on pattern analysis and machine intelligence* 29.6 (2007): 1091-1095.