ABSTRACT

Software effort estimation (SEE) has historically presented a significant challenge for software engineers. The traditional approach estimates the development effort by assigning a score known as a story point, which is measured based on the difficulty level of a user story. Story points are estimated by using methods such as Planning Poker, Dot Voting, the Bucket System, T-shirt Size, etc. These methodologies are heavily dependent on experts, which can lead to inconsistencies and subjective interpretations, ultimately to inaccurate estimations.

Various methods have been proposed to reduce errors and improve model accuracy. Machine learning (ML), deep learning (DL), and large language models (LLMs) are utilized to explore hybrid models that perform better in terms of accuracy and minimize errors. However, inconsistencies in the estimation process remain a major issue. Inconsistencies affect model accuracy, increasing the potential delay of the project. ML and DL models are less capable of capturing the semantic meaning of user story texts. In contrast, the use of LLM in previous studies didn't leverage tokenizer training and instead considered the use of well-written user stories in their practices to improve model accuracy.

This study contributes to training a sub-word wordpiece tokenizer module on the BERT model with the TAWOS dataset to improve the corpus vocabulary for specific terminology domains and implements the well-written format for user stories following the pattern of Role – Actions – Goals (RAG) to improve model accuracy. The evaluation compares MAE, RMSE, Pred(25), and Pred(50) metrics of four benchmark models and performs non-parametric statistical tests to appraise the model consistencies. It yields the highest Pred(25) of accuracy, which is achieved by the BERT-based model that implemented tokenizer training and well-written user stories. This result outperforms the other three benchmark models by approximately 0.30 in the Pred(25) value, or an increase of 36% compared to the original BERT Model.

Keywords: *Software Effort Estimation, story point, effort estimation, tokenizer training, well-written user story, Large Language Model*

INTISARI

Software Effort Estimation (SEE) telah menjadi tantangan jangka panjang bagi para *software engineer*. Pendekatan tradisional mengestimasi beban dan tingkat kesulitan *user story* dengan sebuah nilai yang dikenal dengan *story point* menggunakan metode seperti *planning poker*, *dot voting*, *bucket system*, *t-shirt size* dan sebagainya. Pendekatan tradisional sangat tergantung pada *expert* yang cenderung memprediksi *story point* secara subjektif dan inkonsisten sehingga mengakibatkan estimasi tidak akurat.

Berbagai pendekatan diperkenalkan untuk meningkatkan akurasi dan menekan *error* pada proses estimasi. *Machine Learning* (ML), *Deep Learning* (DL) dan *Large Language Model* (LLM) dieksplorasi untuk mencari model dengan akurasi terbaik. Namun tantangan inkonsistensi pada proses estimasi masih menjadi isu utama. Inkonsistensi estimasi berdampak pada akurasi model sehingga meningkatkan potensi keterlambatan proyek. Model ML dan DL kurang dapat menangkap makna semantik dan kontekstual teks *input*, sementara itu penggunaan LLM pada penelitian-penelitian sebelumnya tidak mengoptimalkan pelatihan *tokenizer* dan mempertimbangkan penggunaan *well-written user story* dalam upaya meningkatkan akurasi model estimasi.

Penelitian ini berkontribusi melakukan *re-training* modul *tokenizer* subkata *wordpiece* pada model BERT dengan dataset TAWOS untuk meningkatkan kosakata *corpus* terhadap domain terminologi yang spesifik dan mengimplementasikan penggunaan *well-written user story* mengikuti pola *Role – Actions – Goals* (RAG) untuk meningkatkan akurasi pada model estimasi. *Output* model utama dan pembanding dievaluasi dengan metrik MAE, RMSE, Pred(25), Pred(50) dan uji statistik non parametrik. Berdasarkan hasil pengujian dan evaluasi didapatkan bahwa model berbasis BERT yang menerapkan pelatihan *tokenizer* dan *well-written user story* memiliki nilai akurasi Pred(25) tertinggi dibandingkan dengan tiga model *benchmark* lainnya yaitu sebesar 0.30. Model tersebut juga mendapatkan peningkatan akurasi tertinggi yaitu sebesar 36%.

Kata Kunci: *Software Effort Estimation*, estimasi, *Agile*, *story point*, *tokenizer training*, *well-written user story*, *Large Language Model*.